ArcIMS® 9

Customizing ArcIMS—Using the ActiveX® Connector

GIS by ESRI™

Contents

Introducing the ActiveX Connector 1

IN THIS CHAPTER

- **About the ActiveX Connector and this guide**
- **Getting started with the ActiveX Connector**
- **About the sample .asp files**
- **Using the ActiveX Connector's ASP template**

The ActiveX® Connector is a component of the ESRI® ArcIMS® application, a scalable, extendable, standards-based software application for distributing mapping and geographic information system (GIS) data and services on the Web.

You can use the ActiveX Connector to develop a custom client application using Visual Basic or the Active Server Pages (ASP) scripting language and to deliver ArcIMS services to that application.

This chapter introduces you to the connector, helps you set it up, and discusses some common tasks demonstrated in the ASP template.

About the ActiveX Connector and this guide

The ActiveX Connector is a component of the ESRI ArcIMS application.

You can use the ActiveX Connector to develop a custom client application using Visual Basic or the Active Server Pages scripting language and to deliver ArcIMS services to that application.

Generally speaking, the job of a connector is to connect your Web server to the ArcIMS Application Server. More specifically, the ActiveX Connector works with custom ActiveX client applications (applications created in ASP or Visual Basic) to provide a quick and easy way to incorporate maps into them.

In addition to basic mapping functionality, such as panning and zooming, you can provide tools to make changes to map layers. Some of the ways you can manipulate a layer include:

- Changing the color or style of a layer
- Adding or removing layers from local data sources
- Updating or adding attribute data

All requests are handled on the server side, which offers two advantages. The generated page is a thin client that is cross-browser compliant. Also, because the processing is on the server side, your code is not exposed to the Internet.

This guide is part of the *Customizing ArcIMS* series of programming references that describes customizing the HTML and Java™ Viewers, creating viewers supported by the ActiveX, Java, and ColdFusion® connectors.

This section discusses authentication considerations.

ActiveX Connector authentication considerations

The ActiveX Connector supports Transmission Control Protocol/Internet Protocol (TCP/IP or TCP) connections as well as Hypertext Transfer Protocol (HTTP) connections. If you want to use authentication, you must use HTTP.

TCP

For a TCP connection, ActiveX Connector communicates with the ArcIMS Application Server directly and is able to access all ArcIMS services running on it.

HTTP

For HTTP connection, the connection chain is:
ActiveX Connector > Web Server >ArcIMS Servlet Connector >ArcIMS Application Server.

On the ArcIMS Servlet Connector, you can set up access control on the ArcIMS services running on the ArcIMS Application Server. To set up the access security of services, configure Esrimap_prop under your servlet engine directory, then change the value for the authenticate property to true. You can put the service access information in one of the following:

- An access control list (ACL) file. Assign it to “aclFileName=..”. There is a sample ACL file, sample_aimsacl.xml, in the ArcIMS Documentation folder. In the following sample ACL file, the SantaClara service is accessible to all users, but the geocode service is accessible only to the user who

has a username of user1 and a password of pass1. Submit the username and password with ArcIMSConnector.Username and ArcIMSConnector.Password properties.

```
<?xml version="1.0"?>
 <AIMSACL>
 <USER name="*" services="SantaClara" />
  <USER name="user1" password="pass1" services="geocode" />
</AIMSACL>
```

- A database. You must edit property values. For example, you must set the useJdbc property to true.

This guide explains the foundation for customizing an ActiveX client application and provides a complete reference to the ActiveX Connector Object Model. It is intended for use by those with a working knowledge of ASP objects and VBScript and familiarity with JavaScript™ and HTML.

Getting started with the ActiveX Connector

The ActiveX Connector and its sample applications are not installed by default when you install ArcIMS, so you must choose to install them during the ArcIMS installation process. The connector comprises a set of dynamic link library (.dll) files. Sample applications for the ActiveX Connector are provided, but you must install them. After you install them, you are ready to connect your custom application to ArcIMS, build more GIS functionality into your custom application, or simply work with the tutorial in this document and sample applications to become familiar with the ActiveX Connector.

The ActiveX Connector comes with the following sample applications:

- A collection of sample .asp files (sample applications) that provide demonstrations of simple mapping functionality.
- An ASP application template to serve as a springboard for developing advanced applications, from the ground up.
- A Visual Basic project to serve as a springboard for developing more advanced applications, from the ground up. It offers a good example of how to incorporate the ActiveX Connector with an existing Visual Basic application. The project references the Aims_activex.dll file to display image URLs from ArcIMS Image Services. Additional documentation for setting up and using this project can be found in <ArcIMS Installation Directory>\ArcIMS\Samples\ActiveX\readme_samples.html.

This section, which helps get you up and running with the ActiveX Connector, is divided into the following topics:

- Installing the ActiveX Connector and its sample applications
- Creating virtual directories for the sample applications
- Starting the SanFrancisco service
- About the sample .asp files
- Using the ActiveX Connector's ASP template

Installing the ActiveX Connector and its sample applications

This procedure describes how to install the ActiveX Connector and its sample applications (.asp files, ASP template, and ArcIMS tutorial data). The sample .asp files referenced in this procedure differ from those used in this guide's tutorial, Chapter 2, 'ActiveX Connector Tutorial'.

1. The ActiveX Connector must be installed on a machine running Internet Information Server (IIS). For development this machine must also have the Microsoft® Internet Explorer (IE) version 5.0.1 or higher installed and a supported integrated development environment (IDE) installed, such as Microsoft Visual InterDev 6.0, SP 3, and Visual Studio 6.0 or .NET. Neither IE nor the IDE must be installed on the production machine. Ensure other aspects of your specific environment and configuration are supported by ArcIMS and its ActiveX Connector by reviewing the ArcIMS system requirements.
   - Go to *http://www.esri.com/software/arcims/index.html*.
   - Click System Requirements.
   - In the resulting page, select the environment options that match your environment, then click Go.
   - In the resulting page, review all the information to ensure that ArcIMS supports your environment and to ensure you're familiar with any limitations or procedures you must perform that are specific to your environment and the ActiveX Connector.

You can review additional system requirements information, especially for Windows 2000 and Windows NT, in Step 1 of the *ArcIMS Installation Guide*. You can access this without reinstalling ArcIMS by navigating to <ArcIMS Installation Directory>\ArcIMS\Documentation\Install.htm. A subsequent step in this procedure tells how to access it from the installation wizard.

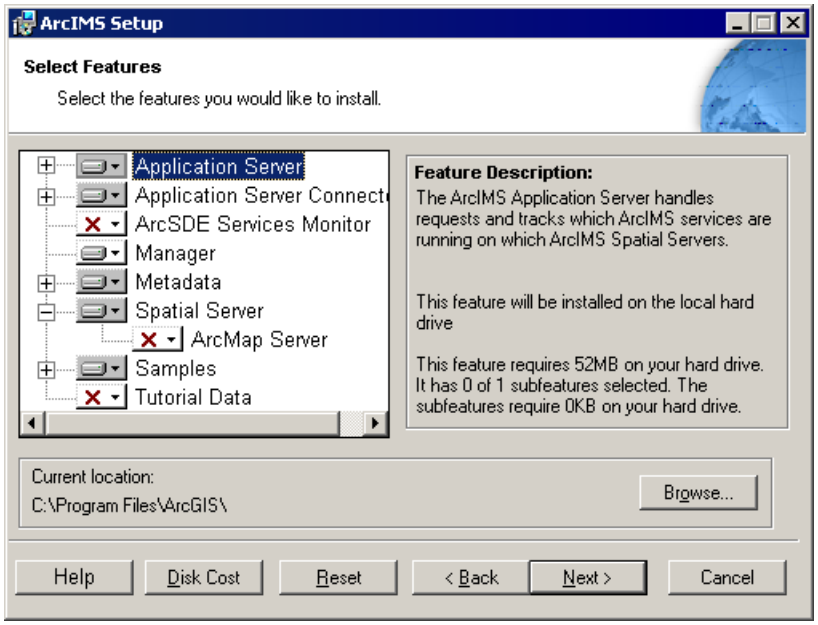
For more information on third party packages that the ActiveX Connector requires, see Knowledge Base article 24124 at *http://support.esri.com/index.cfm?fa=knowledgebase.techarticles.articleShow&d=24124*.

2. Log in to the machine as a user with administrator privileges (full permissions).
3. Ensure that a TEMP variable is set to a folder you have write access to and that has space available.
4. Close all applications on your machine.
5. Insert the ArcIMS installation CD into your CD–ROM drive.

   If your machine has autostart capabilities, it automatically launches the ArcIMS installation wizard. If it doesn't, manually start the wizard:
   - Click Start | Run.
   - Type <drive>:\setup.

   The first panel of the installation wizard appears. If you've already installed ArcIMS, the Add/Remove Application panel displays. If you haven't, the Welcome panel appears.
6. If the Add/Remove Application is the panel that appeared, click Modify, then Next. The Select Features panel appears. Skip to step 8.
7. If the Welcome panel is the one that appeared:
   - Click Next.
   - In the License Agreement panel, read the agreement, then click I accept the License Agreement.
   - Click Next.
   - In the ArcIMS installation panel, click Next.
   - In the ArcIMS Setup panel, click Next.

   The Select Features panel appears.



8. In the Select Features panel, click Help to open the *ArcIMS Installation Guide*. In the guide, review the topic "Installing ArcIMS custom application server connectors" and any additional information on the ActiveX Connector that it links to. To get to this topic you can open the Step 3a booklet in the table of contents or perform a search for the topic title. When finished reviewing the topic, minimize the guide's window. Consult it as needed through the rest of the connector and samples installation.

9. In the Select Features panel, click  to open the Application Server Connectors folder.

10. Click the ActiveX Connector icon  . From the popup menu, select Entire feature will be installed on the local hard drive.

11. Perform the same steps for the Samples folder as you did for the Application Server Connectors folder to install all of the ActiveX Applications and, if you haven't already, all the tutorial data. The ActiveX Connector sample applications use the SanFrancisco.axl configuration file provided in the tutorial data.

12. Click Next through the remaining installation wizard panels and click Finish in the last panel.

    Confirm the installation was successful:
    - Check that the .dll files that make up the connector were installed by navigating to <ArcIMS Installation Directory>\ArcIMS\Connectors\ActiveX. You should see four .dll files: aims_ActiveX.dll, BaseConnector.dll, emaplink.dll, and Util.dll.
    - Check that all sample applications were installed by navigating to <ArcIMS Installation Directory>\ArcIMS\Samples\ActiveX. You should see three folders and a readme_samples.html file. The folder names are ActiveX_Samples, ActiveX_Template, and VB_Demo.

    The ActiveX Connector and its sample applications have been installed.

    If you haven't already used the ActiveX Connector object model to incorporate mapping functionality into your custom application, you'll likely want to become familiar with the ActiveX Connector by using its sample applications (.asp files, ASP template, and ArcIMS tutorial data) and the tutorial in this guide.

    If you intend to use the sample applications, create virtual directories and start the SanFrancisco service. Information on creating virtual directories and starting the SanFrancisco service can be found in this section.

    If you are an experienced programmer or are already familiar with the ActiveX Connector, you are likely ready to begin programming using the object model in Chapter 3 of this guide. Your first steps are to create a connector object, then create a map object and initialize it. After you've performed these steps, all other ActiveX Connector objects in the object model can be referenced.

    As you work with the ActiveX Connector, keep in mind that it is basically a translator between ArcXML tags and the ActiveX Connector object model, so it is helpful to have the ArcXML reference guide open as you work. You can find this at <ArcIMS Installation Directory>\ArcIMS\Documentation\ArcXML_Guide\ArcXML_reference.htm.

For information on how to diagnose ActiveX Connector installation problems, see Knowledge Base article 24005 at *http://support.esri.com/index.cfm?fa=knowledgebase.techArticles.articleShow&d=24005*.

Creating virtual directories for the sample applications

Before you can view the template and sample .asp files you installed, you must create one IIS virtual directory (also known as an alias) for the sample .asp files and one IIS virtual directory for the template.

Virtual directories are directory names that correspond to physical locations (directory paths) on a server. They are used by ArcIMS to serve output to your users and by Web servers to point to physical locations. While similar in concept, ArcIMS virtual directories are different from Web server virtual directories. ArcIMS virtual directories are created automatically. You must create Web server virtual directories in some cases, such as to view the ActiveX Connector sample .asp files.

The following procedure shows how to create Web server (IIS) virtual directories for the sample .asp files and template (instructions are for Windows XP and Windows 2000). When you are ready to create your own custom application, you can use this procedure again to create IIS virtual directories for it.

1. Click Start | Control Panel, double-click Administrative Tools, then double-click Internet Information Services.

2. In the left panel of the Internet Information Services window, click  next to your machine name to open your machine's subfolders.

3. Click  to open the subfolder Web Sites.

4. Under the Web Sites folder, right-click Default Web Site.

5. From the popup menu, select New | Virtual Directory.

   The first panel of the Virtual Directory Creation Wizard appears.

6. Click Next.

7. In the Alias text box of the Virtual Directory Alias panel, type a name for the virtual directory for your sample .asp files, then click Next. It can be any name you choose.

8. In the Web Site Content Directory panel, click Browse, then navigate to <ArcIMS installation directory>\ArcIMS\Samples\ActiveX\ActiveX_Samples, then click Next.

9. In the Access Permissions panel, click Next.

   If you are using IIS 6.0, in the left panel, click  to open the Web Service Extensions folder; then in the right panel, click Active Server Pages, then click Allow.

10. In the completion panel, click Finish.

11. Repeat steps 1 through 10 for the template. For step 8, substitute the following for the path you navigate to: <ArcIMS installation directory>\ArcIMS\Samples\ActiveX\ActiveX_Template.

Starting the SanFrancisco service

Before you can use the template and the sample .asp files you installed, you must create and start an ArcIMS Image Service named SanFrancisco that is based on the SanFrancisco.axl configuration file installed as part of the ArcIMS tutorial data.

1. Start ArcIMS Administrator, then create and start an Image Service named `SanFrancisco` (case sensitive) for the SanFrancisco.axl configuration file. The SanFrancisco.axl file is installed in <ArcIMS Installation Directory>\ArcIMS\Samples\TutorialData\Axl when you install the ArcIMS tutorial data.

   For more information on starting Administrator or creating and starting an Image Service for a .axl file, see the ArcIMS online Help.

2. If you did not accept the default installation location for the ArcIMS tutorial data, which is C:\Program Files\ArcGIS\ArcIMS\Samples\TutorialData, open the SanFrancisco.axl file in a text editor, find the SHAPEWORKSPACE element's LOCATION attribute, and edit it so the filepath points to the directory to which you installed the ArcIMS tutorial data. For example:

   ```
   <SHAPEWORKSPACE name="shp_ws-60"
   directory="My_installation_folder\ArcIMS\Samples\TutorialData" />
   ```

   where My_installation_folder is the path to your ArcIMS installation directory.

   You are ready to begin programming with the ActiveX Connector and to begin using the template and sample .asp files you installed.

About the sample .asp files

The sample .asp files provided with the ActiveX Connector give demonstrations of simple mapping functionality. They can be incorporated as needed into an existing user application. The files, installed to <ArcIMS Installation Directory>\ArcIMS\Samples\ActiveX\ActiveX_Samples, include a default.asp file, which lists all of the sample .asp files. To view the list, in a Web browser, type: `http://<localhost>/ActiveX_samples/default.asp`.

Every sample uses the SanFrancisco service.

Additional documentation for setting up and using these samples can be found in <ArcIMS Installation Directory>\ArcIMS\Samples\ActiveX\readme_samples.html.

Sample 1: Show_Map.asp

Shows a map image and layers list.

Sample 2: Change_layer_visibility.asp

Shows how the visibility of a layer with the layer list can be controlled. The user can check the check box and click Refresh to update the map.

Sample 3: Zoom_In.asp

Shows how the extent of the map can be changed when the user uses the zoom in, zoom out, or pan controls.

Sample 4: Change_layers_order.asp

Shows how the user can change the ordering of the layers by making a layer active, clicking the Up or Down button, then clicking Refresh to update the map.

Sample 5: Change_Marker_Symbol.asp

Sets the size, type, color, outline color, and shadow for a layer. Click Apply to see the changes on the map. The layer must have at least one SimpleRenderer with a SimpleMarkerSymbol.

Sample 6: Change_Polygon_Symbol.asp

Sets the color, style, and interval of the fill and color, width, and style of the outline for a layer. Click Apply to see the changes on the map. The layer must have at least one SimpleRenderer with a SimplePolygonSymbol.

Sample 7: Select_and_Highlight.asp

Shows how a feature is selected and highlighted when a user clicks the feature.

Sample 8: Make_Spatial_Query.asp

Specifies an area to make a selection. Click Refresh to see the selection on the map.

Sample 9: Create_Acetate_Layer.asp

Sets the properties of the North arrow and displays it on the acetate layer of the map.

Sample 10: Query_Attribute_Data.asp

Shows how attribute data can be queried on the active layer. User can perform a query by typing a where clause and clicking Find.

Sample 11: Identify.asp

Shows how feature attribute data can be displayed based on an active layer when the user clicks features in the map.

Sample 12: Buffer.asp

Shows how a user can create a buffer around a feature and select features within a set buffer distance by specifying a buffer distance and clicking Create buffer.

Sample 13: Extract.asp

Shows how features can be extracted from extractable layers to create a shapefile.

Sample 14: Show_Map_http.asp

Shows the map image and layers list accessed by an HTTP connection.

Sample 15: MoveTo.asp

Shows how to use CacheSize and MoveTo in a recordset to control how many records and which records are returned in one ArcXML response from the ArcIMS Spatial Server.

Using the ActiveX Connector's ASP template

The ActiveX Connector's ASP template is a set of .asp, .html, and .gif files that essentially make up a Web site, or viewer, that can serve as a beginning framework for you to develop advanced applications. The template is a more advanced viewer that incorporates many tools and more advanced functionality as well as providing the foundation of an ASP application built with the ActiveX Connector. Functions such as zoom, pan, identify, and query have been incorporated into this template.

Additional documentation for setting up and using this template can be found in <ArcIMS Installation Directory>\ArcIMS\Samples\ActiveX\readme_samples.html.

Opening the template and viewing a list of available services

To open the template, in a Web browser, type:

```
http://<localhost>/ActiveX_Template/index.htm
```

When the template first opens, type the server name and port number where the ArcIMS Application Server is running to get a list of available services.

Setting up the template to use the ActiveX Connector constants

The ActiveX Connector Object Model includes constants that can be used to specify the values of such things as colors or fill styles. See Chapter 3, 'ActiveX Connector Object Model', for a complete listing of the ActiveX Connector constants.

To use these constants in the template, you may need to edit the path in the METADATA TYPE element. The path should point to the location of the aims_ActiveX.dll file in the ArcIMS installation directory. The following example uses the default ArcIMS installation directory path to the .dll file:

```
<!-- METADATA TYPE="TypeLib"
FILE="C:\Program Files\ArcGIS\ArcIMS\Connectors\ActiveX\aims_ActiveX.dll" -->
```

About the template frames

The ASP template is a Web page that contains a set of frames that serve specific mapping purposes. The index.htm file opens the template by defining the frames and specifying the HTML source files for each frame. The six frames that make up the template are TopFrame, ToolFrame, MapFrame, TextFrame, LayerFrame, and BottomFrame.

The TopFrame and BottomFrame are opened by top.htm and bottom.htm to hold static pages displaying a gray background for the Web page. You can change the contents of these HTML files to change the colors, add your logo, or display text.

The other four frames are described below.

ToolFrame

The ToolFrame holds the toolbar for the template. The tools.asp file opens the ToolFrame.

For many of the tools, when a tool is clicked, it is selected but does not perform an immediate action. Instead, the action of the tool starts when an event occurs, such as when the map or a Submit button is clicked. When one of these tools is clicked, setActionState() JavaScript function is called. The setActionState() function passes the value of the tool to MapFrame, where it is held in a hidden input. This ensures that when the user clicks the map, the server knows what action to perform.

When the map is clicked, a form is submitted to map.asp.

MapFrame

The purpose of the MapFrame is to hold the map image and the hidden inputs passed from the ToolFrame. The map.asp file provides content for the MapFrame. Many of the tools are set in the ToolFrame but are executed in the MapFrame.

TextFrame

The TextFrame holds different HTML files, depending on the tool currently in use. When the template is first opened, service.asp opens the TextFrame to show a list of available services. Other files fill the TextFrame to show the results of an identify or the input boxes to create a query.

LayerFrame

The LayerFrame holds the layer list for a service. This layer list is dynamically generated when a service is chosen. The LayerFrame includes three types of functionality: controlling layer visibility, getting layer properties, and setting the active layer. These functions are described later in this chapter. The layers.asp file provides content for the LayerFrame.

ActiveX Connector Tutorial 2

IN THIS CHAPTER

- **Exercise 1: Making a connection to the ArcIMS ApplicationServer**
- **Exercise 2: Retrieving a list of services**
- **Exercise 3: Working with the map**
- **Exercise 4: Working with renderers and symbols**
- **Exercise 5: Working with dynamic layers**
- **Exercise 6: Querying, buffering, and selecting**
- **Exercise 7: Address matching (geocoding)**
- **Exercise 8: Sending and retrieving ArcXML**

This tutorial introduces you to the ActiveX Connector by walking you through some common tasks.

All sample code provided in the tutorial is running in ASP pages and, like most ASP, is a combination of HTML and VBScript.

Before you begin, make sure that the ActiveX Connector is properly installed with ArcIMS. To do so, review the getting started section in Chapter 1 of this guide, 'Introducing the ActiveX Connector'.

In addition to the code examples within the tutorial exercises, each exercise references one or two complete ASP samples. These samples, created specifically for use with the tutorial, are automatically installed in the <ArcIMS Installation Directory>/Samples/ActiveX/ActiveX_Tutorial directory when you install ActiveX Connector. These tutorial samples differ from those referred to in the getting started procedure in Chapter 1. The tutorial samples do not always point to the ArcIMS service SanFrancisco, and some samples, such as the geocoding samples, require extra steps before running.

Exercise 1: Making a connection to the ArcIMS Application Server

In this exercise, you will learn about the elements that make up an ASP header and you will learn how to create a connection to the ArcIMS Application Server. Making a connection is one of the most fundamental tasks you must perform, because without this connection, requests from your ActiveX client application(s) cannot be processed by ArcIMS.

You can connect directly to the Application Server or through your Web server, depending on which type of connection you use, TCP/IP or HTTP, as follows:

- TCP/IP—Use this connection type when the ArcIMS Application Server is on the same local network as the Web server and the ActiveX Connector. To use this connection type, the ArcIMS Application Server must be on the same local network as the Web server and ActiveX Connector.
- HTTP—Use this connection when you want to link to a URL. In this connection, the URL receives the request and sends it to the ArcIMS Servlet Connector, which sends it to the Application Server via your Web server.

The full sample text for this lesson is in connection.asp.

Introduction to ASP

ASP is essentially objects that run within IIS, and they can be created and used within a number of scripting engines.

When you want to add ASP code that communicates with the ActiveX Connector, you use header information to introduce the code. Header information identifies the scripting language that is to follow and typically points to the constants to be used and makes the connection to ArcIMS, as in the following sample of making a TCP/IP connection. Header information is contained in lines 1 through 3. Lines 4 and 16 contain HTML code that is returned to the requesting client.

```
1:    <!--METADATA type="TypeLib" file="C:\Program
       Files\ArcGIS\ArcIMS\Connectors\ActiveX\aims_ActiveX.dll" -->
2:
3:    <%@ Language=VBScript %>
4:    <html><head><title>Connection Example</title></head><body>
5:
6:     <%
7:          Set mConnector = Server.CreateObject("aims.ArcIMSConnector")
8:          mConnector.ServerName="localhost"
9:          mConnector.ServerPort=5300
10:
11:         set services=mConnector.GetServiceNames
12:        for i=1 to services.count
13:                Response.Write "<br>" & services.item(i)
14:         next
15:      %>
16:     </body></html>
```

Line 1 points to the .dll file that contains the constants you want to use, which in this case, are in the ActiveX Connector .dll file. The default installation directory for the .dll is given in this example, C:\Program Files\ArcGIS\ArcIMS\Connectors\ActiveX\aims_ActiveX.dll. By specifying the ActiveX Connector here in the METADATA tag's file attribute, you are setting up the client application so that it can

access constants in the ActiveX Connector, such as `imsGreen` for the color green instead of 65280 for the color green.

Line 3 indicates that the scripting language being used is VBScript. To specify your scripting language, you use an ASP directive. Directives provide global information about the entire page and are always contained inside the starting and ending tags of <%@ and %>. VBScript is the default scripting language for ASP, but you can change this if you want by changing the scripting engine declaration. It will not affect the functionality of ASP or the ActiveX Connector. Anything that falls within the declaration brackets is loaded into memory and executed by IIS when a user accesses the page.

Notice in lines 4 and 16, which contain HTML code that is returned to the requesting client, that the HTML can be interspersed in the ASP code, such as the HTML tag of
 on line 13.

Whenever VBScript is used, it must be placed inside the <% and %> tags as in lines 6 and 15.

Creating a TCP/IP connection

When you create (instantiate) a TCP/IP connection to the Application Server, you create a new object using `aims.ArcIMSConnector` as the input, then you assign the new object to the variable *mConnector*. To create a TCP/IP connection, type the following:

```
Set mConnector = Server.CreateObject("aims.ArcIMSConnector")
mConnector.ServerName="localhost"
mConnector.ServerPort=5300
```

Where:

aims.ArcIMSConnector	is the name of the connector object
localhost	is the name of your host
5300	is the port number of your host

Creating an HTTP connection

When you create an HTTP connection, you use the same connector object as you would for TCP/IP, but for HTTP you specify that it is HTTP and give a URL. The following sample code shows how to create an HTTP connection with authentication. Authentication is optional.

```
Set mConnector = Server.CreateObject("aims.ArcIMSConnector")
mConnector.ConnectUsing=imsConnect.imsConnectHttp
mConnector.ServerUrl = http://localhost
mConnector.UserName="imsuser"
mConnector.Password="ims_user_password"
```

Where:

aims.ArcIMSConnector	is the name of the connector object
imsConnect.imsConnectHttp	indicates that you want to use an HTTP connection type
http://localhost	is the URL your users should use to access your ArcIMS site
imsuser	is your ArcIMS user's username for authentication. This is needed only when the authentication of services is enabled in the ArcIMS Servlet Connector.
ims_user_password	is the password that corresponds to the username specified in imsuser. This is needed only when the authentication of services is enabled in the ArcIMS Servlet Connector.

Exercise 2: Retrieving a list of services

Once you've connected to an ArcIMS Application Server, you can retrieve a list of services that are running on that Application Server.

1. Make the connection to the server. See Exercise 1 for more information.
2. Type the function that retrieves the list of services and assigns the list to the variable `services`:

   ```
   set services=mConnector.GetServiceNames
   ```

3. Process the list by adding the following code:

   ```
   for i=1 to services.count
   Response.Write "<br>" & services.item(i)
   next
   ```

   For each item in the list, the service name is written to the page.

   When the .asp page is run, each service is listed in the HTML page. If you view the source for this page, it will look similar to the following. In this case, three services are listed: "world", "SanFrancisco", and "europe".

   <html><head><title>Connection Example</title></
   head><body>
world
SanFrancisco
europe</body></html>

Exercise 3: Working with the map

Creating a map

The map object is the second most fundamental object, after the connector object, that must be instantiated. The map object lets you access or reference all other ActiveX Connector objects in the object model. Once the map object is created, its InitMap method must be invoked before interaction with an ArcIMS service can occur. Before a map is displayed, an image must be retrieved for the map object.

The InitMap method gets the information about this service, for example, renderers, recordset headers, and geocoding information, and assigns this information to the map object. It has two required arguments: the connector object and the name of the service that will be accessed by the map object. The following code sample links the map object “mMap” to service “SanFrancisco”.

```
Set mMap = Server.CreateObject("aims.Map")
resultInit = mMap.InitMap( mConnector, "SanFrancisco" )
```

After you instantiate the map object, you are ready to get an image to display as the map.

The full sample text for this lesson is in zoomin.asp.

Getting an image to display as the map and change the map properties

The map object has several properties that control the map’s appearance, such as size, map extent, and color. The next code sample changes the map size to 500 x 300 pixels and its background color to red. It also calls the Map.Refresh() method to send this request to ArcIMS. ArcIMS draws the image and puts it in a location accessible through a URL by calling GetImageAsUrl().

```
mMap.Width=500
mMap.Height=300
mMap.BackColor=RGB(255,0,0)
mMap.Refresh
mapurl = mMap.GetImageAsUrl()
```

Navigating

The map object has numerous methods for zooming and panning functions. No matter what methods are used, eventually it’s the extent of the map that is being changed, so you can directly modify the extent of the map object, as in the following code sample.

```
mMap.Extent.XMin = -122.436
mMap.Extent.YMin = 37.7812
mMap.Extent.XMax = -122.43
mMap.Extent.YMax = 37.7866
```

The following ActiveX Connector methods in the map object can simplify the job of modifying the extent.

- ***CenterAt(X as Long, Y as Long)*** requires arguments of the x and y location where the map should be recentered. Note that the x and y coordinates are in pixel units, so you do not have to worry about converting your user’s click to map units before calling CenterAt(). For example:
  `MMap.CenterAt(100,100)`
- ***DoZoom(scaleFactor as Long)*** performs a fixed zoom, in or out. Negative values for scaleFactor result in zooming out, while positive numbers result in zooming in. The larger the absolute value of scaleFactor is, the more the map is zoomed in or out.
  `DoZoomToExtent(envelope as Envelope) : zooms to the extent as defined by the envelope object.`

```
Set env=Server.CreateObject("aims.Envelope")
env.XMax = -121.183
env.XMin = -122.227
env.YMax = 37.5
env.YMin = 36.874
mMap. DoZoomToExtent(env)
```

- ***DoZoomToFullExtent()*** zooms the map to the full extent that is defined in the configuration file. Note that this method does not zoom to the full extent of your datasets.
- ***DoPan (direction as Long, step as Double)*** shifts the current extent to the specified direction. It requires two arguments: the direction to shift and how far to shift, which is called "step". The bigger this value, which should be positive, the farther the shift will be. For example:

  ```
  MMap.DoPan (imsDirection.imsNorth, 5)
  ```

Layers

After the map object has been created and the InitMap method is called, all other ActiveX Connector objects in the object model can be referenced. Two important properties of the map object are the Layers and Legend properties, which enable access to the Layers collection and Legend objects. In fact, the Layers object is a noncreatable object and can be referenced only through the map object.

The Layers object can be thought of as a Layers collection because it is the storage container for the layers (including all three layer types: Feature, Image, and Acetate) in the map display. The Layers object must be accessed via the map object's Layers property, as in the following code sample:

```
Set mLayers = mMap.Layers
Response.Write "There are " & mLayers.count & " layers in the map"
```

There are numerous methods available to the Layers object that enable you to control the positions of layers in the collection, such as add and remove layers and create new layers. The most important function of the Layers object is to provide access to each individual layer, on which the users can perform tasks such as modifying the symbols, performing spatial and nonspatial queries, and buffering. To access a specific layer in the Layers object, use the item() method, which asks for only the index number of this layer. Method find() returns the index number if you know the layer ID. The following code sample shows how to access a layer with ID = "state".

```
set mLayer=mLayers.item(mLayers.find("state"))
Response.Write mLayer.name
```

Exercise 4: Working with renderers and symbols

In this exercise you learn how to render city blocks a certain color based on their population.

Note that renderers and symbols can be used only with Image Services. They are not valid with ArcMap Image Services. The exception is symbols in acetate layers.

Suppose on a layer named `blockgroup`, you want to render the blocks by their population in 1990 (values from field “POP1990”) and also label the blocks by the value, but only at a certain scale range. For blocks with POP1990=1453, you want to render them purple. You want to render 0 to 1452 red, 1545 to 3000 green, and 3001 and above blue. To implement this, you need four polygon symbols, one for each color, and one ValueMapRenderer to organize them.

The full sample text for this lesson is in render.asp.

1. Tell the ValueMapRenderer the name of the field whose value will be used in the rendering. In the following example, the field whose value will be used is POP1990:

   ```
   Set valuerender=Server.CreateObject("aims.ValueMapRenderer")
     valuerender.LookupField="POP1990"
   ```

2. Call method Add() of ValueMapRenderer object to link a purple polygon symbol to information POP1990=1453:

   ```
   Set sps1=Server.CreateObject("aims.SimplePolygonSymbol")
   sps1.FillColor=RGB(255,0,255)
   valuerender.Add sps1,imsExact,1453,0,0
   ```

   More information on the Add() method is provided later in this section.

3. Render blocks with POP1990 from 0 to 1452 red:

   ```
   Set sps2=Server.CreateObject("aims.SimplePolygonSymbol")
     sps2.FillColor=RGB(255,0,0)
   valuerender.Add sps2,1,0,0,1452
   ```

4. To label the blocks with the field POP1990 value, you will need a LabelRenderer, which renders a TextSymbol:

   ```
   Set labelrender=Server.CreateObject("aims.SimpleLabelRenderer")
   labelrender.Field="POP1990"
   Set textsymbol=Server.CreateObject("aims.TextSymbol")
   textsymbol.FontSize=10
   textsymbol.FontColor=RGB(0,0,0)
   labelrender.Symbol=textsymbol
   ```

   In this code sample, LabelRenderer will label each feature with their values in the “POP1990” field. TextSymbol defines how the labels will look.

5. If you don’t want the labels to be drawn on the map all the time, but only display at a certain scale, you will need a ScaleDependentRenderer to wrap the LabelRenderer. ScaleDependentRenderer defines the scale range at which the LabelRenderer will render the TextSymbol:

   ```
   Set scalerender=Server.CreateObject("aims.ScaleDependentRenderer")
   scalerender.Lower=0.00002
   scalerender.Upper=0.0001
   scalerender.Renderer=labelrender
   ```

   The Lower and Upper properties of the ScaleDependentRenderer are the maximum and minimum scales, represented as the number of map units per pixel. Map units per pixel refers to the number of meters, feet, or decimal degrees represented by one pixel in a map. See <SCALEDEPENDENTRENDERER> in the

ArcXML Programmer's Reference Guide for information on the conversion between a relative scale and map units per pixel.

6. Because the blockgroup layer has to take both the ValueMapRenderer and the ScaleDependentRenderer (which already wraps a LabelRenderer), another renderer, GroupRenderer, is needed to wrap these two renderers. Eventually the blockgroup layer will take this GroupRenderer.

```
Set grouprender=Server.CreateObject("aims.GroupRenderer")
grouprender.Add valuerender
grouprender.Add scalerender
mMap.Layers.item(1).Renderer=grouprender
```

Notice that at the initial scale, there are no labels. As you zoom in, labels appear, and as you zoom in past a certain degree, the labels disappear again.

About the Add() method of ValueMapRenderer

Syntax: Add(symbol as Object, range as Long, value as Variant, lower as Variant, upper as Variant)

In this method, the following applies:

- The first parameter, symbol, is the symbol object used for the renderer. The second parameter, range, is an indicator of how to interpret the parameters after it as follows:
  - ImsExact (0) means to render a feature with this symbol if its value is exactly the same as value.
  - imsRange.imsRange (1) means to render features with this symbol if its value is within the range of lower and upper. (Note: you have to use either 1 or imsRange.imsRange, because a single imsRange will cause an error since it's both the name of this constant group and this constant).
  - ImsOther (2) means to render a feature with this symbol if its value is other than the values defined previously by imsExact or imsRange.
- The third parameter, value, is the exact value used when the second parameter is imsExact; otherwise, set it to 0. When imsOther is used in the first parameter, set "value" to 0.
- The fourth and fifth parameters, lower and upper, are the lower and upper range of the values used when the second parameter is imsRange. Otherwise, set them to 0. When imsOther is used in the first parameter, set parameters lower and upper to 0.
- Parameters value, lower, and upper are of type Variant since they can be of any type, depending on the type of the lookup field.

Exercise 5: Working with dynamic layers

Preparing the ArcIMS service

You can dynamically add an acetate, feature, or image layer to a map. Before adding a feature or image layer, you must make sure that the service accepts dynamic layers. In the configuration file, the "dynamic" attribute of the <MAP> tag should be set to "true". For example, if you are using the map service "SanFrancisco", open SanFrancisco.axl in a text editor, find the <MAP> tag, and add dynamic="true":

```
<MAP dynamic="true">
```

Save the configuration file, refresh the service in Administrator, and save the Administrator site configuration. Now the service is ready to accept feature layers and image layers other than the layers defined in the configuration file.

Adding an acetate layer

On an acetate layer, you can add points, lines, polygons, North arrows, scalebars, and text. Adding an acetate layer is the same as adding any other layer to the collection.

1. Create an acetate layer with its default visibility being true:

   ```
   Set acetatelayer= Server.CreateObject("aims.AcetateLayer")
   ```

2. Create the objects you want to add to the acetate layer and set their size, color, style, and position on the map.
3. Add all the objects to the acetate layer.
4. Add the acetate layer to the Layers object of the map object, and call Map.Refresh().

Adding a feature layer

Adding a feature layer is valid only with Image Services and not with ArcMap Image Services.

For a feature layer, you will first need to build a workspace; build a ShapeWorkspace if your data is from a shapefile, but use an SDEWorkspace for ArcSDE data. A workspace contains the information of the data used in the new layer, the directory where the files are, the filename, and the shape of the feature: point, line, or polygon.

```
Set mShapeWorkspace = Server.CreateObject("aims.ShapeWorkspace")
mShapeWorkspace.Directory = "C:\Program
Files\ArcGIS\ArcIMS\Samples\TutorialData\SanFrancisco"
mShapeWorkspace.FeatureClass = imsFeatureClass.imsPoint
mShapeWorkspace.Name = "artgalleries"
```

The Create() method of the Layers object takes the workspace information—it will look for shapefiles named "artgalleries" under the workspace directory—and creates a point layer. The next step is to add the new layer to the Layers object. You can also set a new symbol to the new layer if you don't like the default symbols.

```
Set fLayer = mMap.Layers.Create(mShapeWorkSpace)
mMap.Layers.add fLayer
```

Adding an image layer

Adding an image layer is similar to adding a feature layer. You will first build an ImageWorkspace with the information of the image file, then call the Create() method of the Layers object. Adding an image layer is valid only with Image Services and not with ArcMap Image Services.

The full sample text for this lesson is in imagelayer.asp.

Exercise 6: Querying, buffering, and selecting

Often when you perform query or buffer functions, you'll want to follow them by displaying, selecting, or highlighting the results of the query or buffer. This exercise describes these tasks.

Spatial query

Point-based, on a Polygon layer

The most commonly known spatial query is likely one in which the user clicks on a map to identify the feature being clicked. The logic behind this operation requires you to:

1. Create a point object from the mouse click.
2. Add this point as a spatial filter to the Filter object of the layer.

   Only features that contain the point will be drawn on the map; other features are "filtered out".

   Note that a layer can have a filter, and its recordset can also have a filter. The layer's filter determines which features are drawn on the map, and the recordset's filter determines which records are stored in the Recordset object. The filter for the layer will not be automatically applied to the recordset. If you want to filter out records, you must also apply the filter on the recordset.

Envelope-based

Using a point as the spatial filter on a Polygon layer works well, but for Line and Point layers, it's difficult to click right on a point or on a layer. One solution is to build a small envelope around the point and use this envelope as the spatial filter so all the features that fall within or touch the envelope will be selected. The following sample shows how to build an envelope around the mouse click and set it as the spatial filter. Notice in this sample that if you didn't click a store "on", none of the stores will be displayed, so you may want to highlight the selected features instead of filtering out the unselected features. For more information, see 'Highlighting the selected features' in this section.

```
   Set mLayer = mMap.Layers.item(6) 'point layer

   if Request.QueryString("map.x")<>"" then
      Set env=Server.CreateObject("aims.Envelope")
      set point=mMap.ToMapPoint(
clng(Request.QueryString("map.x")),clng(Request.QueryString("map.y")))
      r=(mMap.Extent.XMax-mMap.Extent.Xmin) /30
      env.XMax=point.x+r
      env.XMin=point.x-r
      env.YMax=point.y+r
      env.YMin=point.y-r
      mLayer.Filter.ClearGObjects
      mLayer.Filter.AddGObject env
      mLayer.Recordset.Filter.ClearGObjects
      mLayer.Recordset.Filter.AddGObject env
      Response.Write "Records selected: " & mLayer.Recordset.count
      if mLayer.Recordset.movefirst then
      do
        Response.Write "<br>Store Name=" &
mLayer.Recordset.Fields.FieldValueAsString(2)
      loop while mLayer.Recordset.movenext
      end if
   end if
   mMap.Refresh
```

Nonspatial query

A nonspatial query, or attribute query, is a query on the attributes of the layer. Instead of setting a spatial filter, a nonspatial query sets a SQL “where” clause to the filter’s WhereExpression field, and only the features with attributes satisfying the SQL clause will be drawn, as demonstrated in the following sample.

```
Set mLayer = mMap.Layers.item(1) 'polygon layer

   mLayer.Filter.ClearGObjects
   mLayer.Filter.WhereExpression="POP1990  >=  4531"
   mLayer.Recordset.Filter.ClearGObjects
   mLayer.Recordset.Filter.WhereExpression = "POP1990  >=  4531"
   Response.Write mLayer.Recordset.count & " Records with POP1990  >=  4531"

   mMap.Refresh
   mapurl = mMap.GetImageAsUrl()
```

Highlighting the selected features

The previous query samples display only the selected features. Two techniques are available for displaying all the features and highlighting the selected ones.

The first technique is to clone the layer that is the target of the selection and to modify its symbol at the same time a filter is applied. When a layer is cloned, it retains all of the properties of the original layer, including its symbol. You can change the symbol to a color to indicate features have been selected. In the following sample, notice that the last layer is the clone of the first layer:

```
   Set mLayer = mMap.Layers.item(4).clone 'polygon layer
   mMap.Layers.add mLayer
   mLayer.Filter.ClearGObjects
   set env=Server.CreateObject("aims.Envelope")
   env.XMin=-121.030023480025
   env.YMin=37.1076155165431
   env.XMax=-121.989991947739
   env.YMax=37.8015226693018
   mLayer.Filter.AddGObject env
   mLayer.Recordset.Filter.ClearGObjects
   mLayer.Recordset.Filter.AddGObject env

   set renderer = Server.CreateObject("aims.SimpleRenderer")
   select case mLayer.featureclass
      case "point"
        set    pointSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
        pointSymbol.Color = 65535 'yellow
        pointSymbol.Width=8
        renderer.Symbol=pointSymbol

      case "line"
        set lineSymbol = Server.CreateObject("aims.SimpleLineSymbol")
        lineSymbol.Color = 65535
        lineSymbol.Width=4
        renderer.Symbol = lineSymbol
      case "polygon"
        set polySymbol = Server.CreateObject("aims.SimplePolygonSymbol")
        polySymbol.Fillcolor = 65535
        renderer.Symbol = polySymbol
```

```
end select
mLayer.Renderer = renderer
mMap.DoZoomToExtent(env)
mMap.Refresh
mapurl = mMap.GetImageAsUrl()
Response.Write "<BR>The Layers collection is:"
 for i=1 to mMap.Layers.count
  Response.Write "<br>Layer Name=" & mMap.Layers.item(i).name
  Response.Write ", Layer ID=" & mMap.Layers.item(i).ID
 next
```

The second technique uses the CreateHighlight method, which creates a new feature class layer that contains specified objects only. The recordset of the new layer will contain only highlighted objects. By default, a renderer is specified (50 percent transparent yellow). This method applies only to nonspatial queries. A sample of using the CreateHighlight method follows:

```
Set mLayer = mMap.Layers.item(1)
   Set highlight=mMap.Layers.CreateHighlight(mLayer,"POP1990","4531,4715,4737")
   mMap.Layers.add highlight
   Response.Write "Records highlighted : " & highlight.Recordset.count

   mMap.Refresh
   mapurl = mMap.GetImageAsUrl()
   Response.Write "<BR>The Layers collection is:"
    for i=1 to mMap.Layers.count
     Response.Write "<br>Layer Name=" & mMap.Layers.item(i).name
     Response.Write ", Layer ID=" & mMap.Layers.item(i).ID
    next
```

Buffering

Buffering is the creation of a polygon around a geographic feature (point, line, or polygon) based on a given distance, and this polygon is then used as the spatial filter. “How many customers are within one mile of my store?” is an example of a question that can be solved through buffering.

Adding functionality to enable buffering involves writing code using a number of different objects. Before writing this code, it is important to have an understanding of the three layers involved in the buffering:

- The first layer is the original layer, the layer containing the geographic features that are the center of the buffer. In the example “How many customers are within one mile of my store?” the store layer is the original layer.
- The second layer is the target layer, the layer containing the buffer layer (the features falling within the buffer)—for example, the customer layer.
- The third layer has no geographic features on it; it’s just a layer that includes the buffer image.

The process of buffering is the combination of two queries. The first query is on the original layer and can be spatial or nonspatial to select the features to buffer. A buffer is then drawn around the selected features on the original layer. After the buffer is drawn, the second query is applied—a spatial query with the buffer as the spatial filter on the target layer—to select features that fall within the buffer.

The specific steps to perform this procedure follow.

Step 1: Make clones for the original and target layers, assign special symbols to them (or they will use the symbols in the configuration file), then add them to the Layers collection. See the following code sample for preparing the original layer.

```
Set origLayer = mMap.Layers.Item(6).Clone()
origLayer.Name = "my_store_Layer"
set sms1=Server.CreateObject("aims.SimpleMarkerSymbol") sms1.Color=RGB(0,255,0)
sms1.Width=20
set sr1 = server.CreateObject("aims.SimpleRenderer")
sr1.symbol = sms1
origLayer.Renderer=sr1
mMap.Layers.Add origLayer
```

Step 2: Set up a filter on the original layer. This filter has two functions. One is to query the features on the original layer, “my store”, as in the following code sample.

```
origLayer.Filter.WhereExpression="STORE_ID = '1'"
origLayer.Recordset.Filter.WhereExpression="STORE_ID = '1'"
```

The other function is to establish the conditions for the buffer. You use the Buffer and BufferUnits properties on the Filter object to determine the buffer distance and units, as in the following code sample.

```
origLayer.Filter.BufferUnits = imsMiles
origLayer.Filter.Buffer = 0.1
```

Step 3: Invoke the buffering process by using the CreateBuffer() method on the map’s Layers object. CreateBuffer() returns the third layer, the buffer layer for the buffer image. Add this layer to the Layers collection to display, as shown in the following sample:

```
Set bufferLayer = mLayers.CreateBuffer(origLayer.Filter, targetLayer)
bufferLayer.name="buffer_image_layer"
mLayers.Add bufferlayer
```

A new layer containing the buffer image is created and has a 50 percent transparent yellow fill by default. You can change the appearance of the buffer layer by accessing the Renderer properties of the buffer layer’s symbol object. By default, a buffer layer uses a SimplePolygonSymbol object as a Renderer, but you can specify other Renderer objects to change the buffer’s appearance.

Step 4: Assign a unique layer ID to the three layers if necessary. Remember to do this step after the Map.Refresh(). Set “loadRenderer” and “loadRecordset” to true in initMap; otherwise, the buffer request will not be correctly built, and the table header information (Recordset->TableDesc) will be empty when displaying the records.

Note that if you don’t assign a unique layer ID to the clones of the original layer, target layer, and buffer layer, the ActiveX Connector assigns them the same layer ID.

Exercise 7: Address matching (geocoding)

Address matching, or geocoding, is a process that compares components of an address you enter to a street database to determine whether or not there is a match. If there is a match, a score indicates how close the match is. Typically after a match is made and found to be a close match, you'll want to display the address on the map.

Address matching works with Image Services only (not ArcMap Image Services).

To perform address matching, you must have a street file that contains street centerline segments with address information. A street file can be a shapefile or an ArcSDE layer. Shapefiles and geocoding indexes are stored in the same directory. The geocoding indexes for ArcSDE layers are stored in C:\TEMP on Windows and /tmp on UNIX by default. However, ArcSDE index files can be stored in any location. Before you begin address matching, you must set the properties for the street file you are geocoding against. You must also build a geocoding index.

This exercise describes how AddressMatchInputs works and walks you through the following tasks:

- Setting the properties for the street file (preparing a map service for address matching)
- Building a geocoding index
- Sending a request to start the match checking
- Drawing an address location on the map

Setting properties for the street file and building a geocoding index

When you set properties for the street data layer, you can either:

- Edit the existing configuration file, for example, the SanFrancisco.axl file used in the sample applications.
- Create a new configuration file from the street file.

The following procedure creates a new configuration file from the street shapefile from the SanFrancisco sample in the ArcIMS tutorial data. Note that if you decide to edit the existing configuration file and change the address style in it, you must refresh the corresponding ArcIMS service.

1. Start up Author. For instructions on starting up Author, see the ArcIMS Help.
2. In Author, click the Add Layers button  .
3. In the Catalog window, navigate to the streets shapefile, which is located in the <ArcIMS Installation Directory>\ArcIMS\Samples\TutorialData\SanFrancisco folder. For more information on using the Catalog window, see the ArcIMS online Help.
4. Click the file, click the Add Layers button  , then click the Close button  .
5. In Author, click the street layer in the legend to make it active.
6. Click the Geocoding Properties button.
7. Click the Address Style dropdown arrow, and click US Streets with Zone.

   The address style you set here determines the content of the AddressMatchInputs. For more information on how the style determines the content, see 'How AddressMatchInputs works in the ActiveX Connector' in this chapter.

For descriptions of each address style and general information on styles, see the <GCSTYLE> and <GCTAG> tag descriptions in the *ArcXML Programmer's Reference Guide*.

8. Click a cell in the Field column, then click a field on the dropdown list that appears. You must assign fields to Values with an asterisk. If ArcIMS detects an appropriate field, it populates the Field column with it.
9. Check Build Geocoding Index.
10. Click OK.

    If you are geocoding a layer that is not an ArcSDE layer, ArcIMS builds the geocoding index and closes the dialog box. This signals that you are finished setting properties for the street file and building a geocoding index, and you can skip to Step 13.

    If you are geocoding an ArcSDE layer, the Geocoding Index Directory dialog box appears.
11. In the Geocoding Index Directory dialog box, type the name of a directory to which you want to save the geocoding index. The default directory is C:\TEMP on Windows or /tmp on UNIX.
12. Click OK.

    ArcIMS builds your geocoding index.
13. Save the configuration file as geocode.axl to a directory you choose.

    To check that ArcIMS saved your geocoding index information, open the geocode.axl file you just created in a text editor. You should see the following lines of code:

```
<EXTENSION type="Geocode">
<GCSTYLE name="USAddressZ">
<GCFIELD id="FromLeft" name="L_F_ADD" />
<GCFIELD id="FromRight" name="R_F_ADD" />
.....
</GCSTYLE>
</EXTENSION>
```

14. Create and start an ArcIMS service based on the geocode.axl file you just created. Name the service geocode.

    Your new service is now ready to accept geocoding requests from your custom application.

How AddressMatchInputs works in the ActiveX Connector

AddressMatchInputs takes an address and builds a geocoding request. It is helpful to understand how AddressMatchInputs works by imagining a table called AddressMatchInputs that stores address information. In this table, a full address—for example, one that includes street name, city, state, ZIP, and cross street—is broken down piece by piece into individual records. One record becomes `street name` and another record becomes `city`, and so on, but each record is still part of the full address.

Each record, or part of an address, has five fields to describe it:

- ID
- Description
- Label
- Width
- Value

For example, the address 850 Balboa St with ZIP Code 94118, when in the US Streets with Zone address style, is represented as:

ID	Description	Label	Width	Value
STREET	street number, name, and type	Street	10	850 Balboa St
ZONE	zone information	Zone	5	94118
CROSSSTREET	cross street name and type	Cross street	10	

The first four fields are populated with the information from the ArcIMS Spatial Server, but you must fill in the fifth field, the Value field. For more information on how the first four fields are populated, see the description for <GCINPUT> in the *ArcXML Programmer's Reference Guide*.

The number of records in this table and the ID for each row depend on the address style you picked while building the geocoding index. If you chose US Streets with Zone, then there should be three records, with IDs "STREET", "ZONE", and "CROSSSTREET".

Run the sample code in addressstyle.asp and see how the above information is displayed in a table. The Address style is shortened and saved in the property Style of AddressMatchInputs. Notice the Value field is empty. For the address style "US Streets with Zone", or "USAddressZ", you must provide a value for "STREET" and "ZONE". A value for "CROSSSTREET" is optional. See the description in the *ArcXML Programmer's Reference Guide* of <GCTAG> for information on other address styles. If you change the address style in the configuration file, refresh the map service and run sample addressstyle.asp again to change the records in the table.

Finding 850 Balboa St with ZIP Code 94118 on the map

To find address "850 BALBOA ST" with ZIP Code 94118 on the map, loop through AddressMatchInputs, then set "850 BALBOA ST" to the "STREET" record and "94118" to the "ZONE" record. If you know the cross street, set it to record "CROSSSTREET". Sample code follows.

```
Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
if mAddressMatchInputs.MoveFirst() then
do
if mAddressMatchInputs.Id = "STREET" then
mAddressMatchInputs.Value = "850 BAL ST"
end if
if mAddressMatchInputs.Id = "ZONE" then
mAddressMatchInputs.Value = 94118
end if
loop while mAddressMatchInputs.MoveNext()
end if
```

Next, call the method AddressMatch() to send the request. The Spatial Server processes this address against the street information and finds points that match this address. Each point is a candidate and has a score indicating how well it matches this address. In the two parameters of method AddressMatch(), you can control the maximum number of candidates the Spatial Server will return and the minimum score each candidate must pass.

Drawing an address on the map

After the AddressMatch() request is sent, the candidate points are returned with a location in AddressMatchResult, each having a corresponding score. A common way to draw an address on the map is to find the point with the highest score (usually the first one) and draw a point on an acetate layer. When adding the point on an acetate layer, keep in mind that the unit of the point's coordination is not in pixels but in the same units as the map. In the AcetateLayer.Add() method, the second parameter should be imsDatabase. For a complete code sample, see addressmatch.asp.

Exercise 8: Sending and retrieving ArcXML

No matter which objects and methods have been called, eventually the job of the ActiveX Connector is to send an ArcXML request to the Application Server and to parse the ArcXML response. ArcIMSConnector is the class that does the communication. Its property RequestAXL and ResponseAXL contain the last ArcXML request sent from the connector and the ArcXML response from the Application Server. If there is an ERROR message in the ArcXML response, its content can be accessed by calling method ArcIMSConnector.GetLastError().

If you need to perform some functions that are not in the ActiveX Connector, you can build the ArcXML request and send it directly to ArcIMS via the SendAxlRequest() method. The following two lines actually send the same request to the Application Server.

```
mConnector.SendAxlRequest "catalog", "<GETCLIENTSERVICES/>"

Set services=mConnector.GetServiceNames
```

The first line sends its own ArcXML request directly. From the syntax of ArcIMSConnector.SendAxlRequest, the first parameter should be the name of an ArcIMS service; however, in the above code snippet, “catalog” is not the name of a service but a keyword for querying the information of all the services.

The second line calls the method GetServiceNames(), which will build a request and send it. If you choose to send ArcXML requests directly, you are responsible for parsing the ArcXML response.

ActiveX Connector Object Model

3

IN THIS CHAPTER

- **ActiveX Connector objects**
- **ActiveX Connector methods, properties, and constants in alphabetical order**

This chapter describes the objects, methods, properties, and constants of the ActiveX Connector Object Model.

The ActiveX Connector Object Model is highlighted by two key objects that act as gateways to the rest of the objects in the model.

The ArcIMSConnector object must be created before any other object can be accessed. By setting its ServerName and ServerPort properties, a connection is established with the ArcIMS Application Server. At this point, there is no association with a service.

The Map object is the second critical object to be created. Once the Map object is created, the InitMap method must be invoked before interaction with a service can take place. The InitMap method has two required arguments: the ArcIMSConnector object and the name of a service that will be accessed by the map.

Once the map and ArcIMSConnector objects have been instantiated, all other objects in the model can be referenced.

ActiveX Connector objects

AcetateLayer

An AcetateLayer object defines a layer that is displayed on top of the map image. ScaleBarObjects and NorthArrowObjects, as well as simple graphics, such as LineObjects, PointObjects, and PolygonObjects, are typically added to this layer. A map may contain multiple acetate layers.

AddressMatchInputs

The AddressMatchInputs object is a property that contains all of the street address and ZIP Code information necessary to successfully geocode an address on a geocodable map FeatureLayer. Typically, the FeatureLayer represents a street network with address attributes. The AddressMatchInputs object is noncreatable.

AddressMatchResults

The AddressMatchResults object is a property of the AddressMatchInputs object and contains the actual address value, address matching Score, and PointObject for each candidate that was successfully returned from a geocode request. The AddressMatchResults object is noncreatable.

ArcIMSConnector

The ArcIMSConnector object represents a persistent connection to an ArcIMS Spatial Server. This object is used to initialize a Map object and establish a relationship between that Map object and an ArcIMS service. The ArcIMSConnector object and the associated Map object are the gateways through which all the other ActiveX Connector objects are created.

CalloutMarkerSymbol

The CalloutMarkerSymbol object will display a label and a callout box around the label. This symbol can only be used with point features. It can be rendered with special effects, such as glowing, shadowing, outlining, and antialiasing.

Circle

The Circle object, like the Envelope object, is a spatial representation object that is used to define a spatial filter for features in a FeatureLayer.

Envelope

The Envelope object represents the rectangular extent of geographic features. Much like the Circle object, it can also be used to define a spatial filter for features in a FeatureLayer. The Envelope object also represents the entire geographic extent of a FeatureLayer.

FeatureLayer

The FeatureLayer object represents all of the geographic features in a dataset that are contained in a single service layer. The valid feature classes contained in a service's FeatureLayer object are points, lines, and polygons.

Fields

The Fields object is a collection of database table fields contained in a Recordset object.

Filter

The Filter object is a property of the FeatureLayer object. It contains all the spatial and attribute constraints used to filter out a subset of features from the FeatureLayer. The Filter object is also a property of the Recordset object and will filter out a subset of records from the recordset.

GradientFillSymbol

The GradientFillSymbol object will fill a polygon feature with a gradual variation of color, ranging from a start color's value to a finish color's value.

GroupRenderer

The GroupRenderer object represents a collection of renderer objects that are used together to display FeatureLayer information with more complex symbology. A FeatureLayer can have a GroupRenderer associated with it that contains any number of different renderer types.

HashLineSymbol

The HashLineSymbol is used to render linear features using two different line styles. The two line styles represent a background and foreground symbol. One use for the HashLineSymbol would be to display railroad features.

ImageLayer

The ImageLayer object represents an image file that is displayed as a layer in a Service. It is noncreatable.

ImageWorkspace

The ImageWorkspace object provides a mechanism for locating image files that are to be added to a service as ImageLayers. The Directory property contains the system directory where the image file is located, and the File property contains the name of the image file that is the source for the ImageLayer. The new ImageLayer is then created by invoking the Create method in the Map object, which takes the ImageWorkspace object as the method parameter.

Layers

The Layers object is a property of the Map object. It is a collection of layer objects in a map service. The Layers collection can contain any combination of FeatureLayers, ImageLayers, and AcetateLayers.

Legend

The Legend object represents a table of contents for the layers contained in a service. The Legend object is a property of the Map object, and each Map object has one legend. The Legend object is used to create a cartographically pleasing image of a map service's legend that can be retrieved and placed into a Web page or a Windows-based application.

LineObject

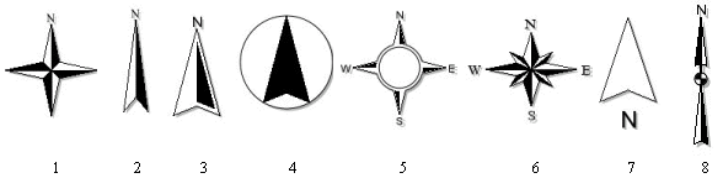
The LineObject represents a linear graphic that can be added to an AcetateLayer. The LineObject can be displayed on an acetate layer with any of the same symbol properties available to features in a FeatureLayer.

Map

The Map object contains all the information that can be obtained from a specified service, such as information about each layer, the map extent, and the map units. The Map object is not usable until its InitMap method is invoked, which requires an ArcIMSConnector and a service name. This method will establish a relationship between the Map object and a service. Like the ArcIMSConnector, the Map object is a kind of parent object from which all the other ActiveX Connector objects are created.

NorthArrowObject

The NorthArrowObject is a graphic object that can be added to an AcetateLayer to indicate the orientation of the map display. There are eight different North arrow styles that can be used.



Parts

The Parts object is a collection of Points objects (see Points) that define the geometry of PolygonObjects and LineObjects. Each part of a Parts collection is a Points object collection representing one polygon for a PolygonObject or one line segment for a LineObject.

PointObject

The PointObject is an x,y location object. Generally, a PointObject is added to and rendered in an AcetateLayer. It can also be added to a Filter object and used as a spatial filter. When using the AddressMatching objects, a PointObject is returned from an AddressMatchResults object.

Points

The Points object is a collection of points. The Points object can be used in Line, Polygon, and Parts objects.

PolygonObject

The PolygonObject represents a polygon shape that can be rendered on an AcetateLayer. The PolygonObject has a Parts property, which contains points that represent the coordinates for the polygon boundary.

RasterFillSymbol

The RasterFillSymbol object contains properties that fill a polygon symbol with a specified raster image.

RasterMarkerSymbol

The RasterMarkerSymbol object contains properties that symbolize point features with a specified raster image.

RasterShieldSymbol

A RasterShieldSymbol object is a user-specified image that is used as a custom shield to identify roads (or other line features). The text associated with the image comes from a specified field and is placed on top of the image.

Recordset

The Recordset object contains all records for a specified feature layer. Each record in a recordset represents one feature in a feature layer. The Recordset object is noncreatable.

ScaleBarObject

The ScaleBarObject represents the scalebar graphic that indicates the scale of a map. The object is added to and positioned on an AcetateLayer before it can be displayed on a map image. The ScaleBarObject is positioned on a map's AcetateLayer by using its x- and y-values.

ScaleDependentRenderer

The ScaleDependentRenderer object displays specified rendering information at certain scales. A layer can have different renderings depending on the scale threshold. For example, when zoomed out, you can draw a street layer one pixel in width. As you zoom in, you can draw the street layer eight pixels in width and in a different color.

SDEWorkspace

The SDEWorkspace object defines a data source stored in an ArcSDE database. It can be passed to the Create method of the Layers collection to dynamically create new ArcSDE layers for services.

ShapeWorkspace

The ShapeWorkspace object defines a shapefile data source from which shapefiles can be specified and added dynamically to a running map service as a layer.

ShieldSymbol

The ShieldSymbol object is used to add simple roadway shield symbols. The ShieldType property lets a developer choose from five shield options—Interstate, USRoads, Rectangle, Oval, and Mexican.

SimpleLabelRenderer

The SimpleLabelRenderer object is used for labeling features in a feature layer. A field property is used to specify the name of the field from which values will be derived for labeling. Other properties, such as FWeight, LWeight, and LabelPriorities, are available for considering feature weights, how many features to label, and label rotation and positioning.

SimpleLineSymbol

The SimpleLineSymbol object is used to render line features in a feature layer. Properties, such as color, style, and width, are used to set a line feature's symbology.

SimpleMarkerSymbol

The SimpleMarkerSymbol object is used to symbolize point features using one of the five predefined symbol types: circle, triangle, square, cross, or star.

SimplePolygonSymbol

The SimplePolygonSymbol object is used to symbolize polygon features in a feature layer. Properties, such as FillColor, FillStyle, and FillInterval, are used to set a polygon feature's symbology.

SimpleRenderer

The SimpleRenderer object is used to display features (point, line, polygon) using one symbol. Layers with a SimpleRenderer will display all features with the same symbology.

TableDesc

The TableDesc object contains all of a recordset's field information (name, type, length, and precision).

TextMarkerSymbol

The TextMarkerSymbol object is used to set the appearance of text created by the TextObject. Properties, such as Font, FontColor, FontStyle, and FontSize, alter the way the text is displayed.

TextObject

The TextObject is used to place text on an acetate layer (created by an AcetateLayer object) to be displayed on the map.

TextSymbol

The TextSymbol object is used to label point, line, and polygon features. Properties, such as Font, FontColor, FontStyle, and FontSize, change the feature's labeling.

TrueTypeMarkerSymbol

The TrueTypeMarkerSymbol object defines the characteristics of labels associated with a map layer using a TrueType® font. Using a TrueType font allows you to use scalable vector fonts, which can be scaled to any size and otherwise transformed more easily than a bit map font and with more attractive results.

ValueMapLabelRenderer

A ValueMapLabelRenderer object provides a way to label the features of a map layer by drawing a Symbol for each unique data value or range of data values specified by the Value property. The LookupField property is the name of the field in the layer that stores the text values to use as labels. The Symbol property defines how the text is drawn on the feature. The ValueMapLabelRenderer supports both unique values and ranges.

ValueMapRenderer

The ValueMapRenderer object provides a way to symbolize features of a map layer by drawing a Symbol for each unique data value or range of data values. The LookupField property is the name of the field in the layer that stores the text values to use. Depending on the type of feature, the Symbol property defines how the feature is drawn on the map.

AcetateLayer.Add method

Description

Adds an object—for example, a point, line, or North arrow—to the acetate layer.

Category

AcetateLayer

Syntax

AcetateLayer.Add(object as Object, units as imsUnit) as Boolean

Arguments

object	Object to add to the acetate layer.
units	Determines how coordinates of object are specified. 0 = pixel units; 1 = database units.

Returned Value

Boolean	True indicates that the object was added to the acetate layer. False indicates that an error occurred and that the object was not added to the acetate layer.

Example

```
Dim mAcetateLayer
Dim mPointObject
Set mAcetateLayer = Server.CreateObject("aims.AcetateLayer")
mAcetateLayer.visible = true
Set mPointObject = Server.CreateObject("aims.PointObject")
mPointObject.x = -117.00
mPointObject.y = 45.00
mAcetateLayer.Add mPointObject, 1
mapimage.Layers.Add mAcetateLayer
mapimage.refresh
```

See Also

AcetateLayer.Clear
AcetateLayer.Item
AcetateLayer.Remove

AcetateLayer.Clear method

Description
Clears all objects from the acetate layer.

Syntax
AcetateLayer.Clear()

Arguments
None

Returned Value
None

Example
```
Dim mAcetateLayer
Dim mWrongPoint
Dim mRightPoint

Set mAcetateLayer = Server.CreateObject("aims.AcetateLayer")
mAcetateLayer.visible = true
Set mWrongPoint = Server.CreateObject("aims.PointObject")
mWrongPoint.x = -117.0
mWrongPoint.y = 45.0
mAcetateLayer.Add mWrongPoint 1
mAcetateLayer.Clear

Set mRightPoint = Server.CreateObject("aims.PointObject")
mRightPoint.x = -100.0
mRightPoint.y = 55.0
mAcetateLayer.Add mRightPoint 1
mapimage.Layers.Add mAcetateLayer
mapimage.refresh
```

See Also
AcetateLayer.Add
AcetateLayer.Item
AcetateLayer.Remove

AcetateLayer.Count property

Description
Number of objects in the acetate layer. Read only.

Type
Long

Example
```
Dim mAcetateLayer
Dim mPointObject

Set mAcetateLayer = Server.CreateObject("aims.AcetateLayer")
mAcetateLayer.visible = true

Set mPointObject = Server.CreateObject("aims.PointObject")
mPointObject.x = -117.0
mPointObject.y = 45.0

mAcetateLayer.Add mPointObject 1
mCount = mAcetateLayer.Count

Response.write mCount
```

See Also
AcetateLayer.Add
AcetateLayer.Clear
AcetateLayer.Remove

AcetateLayer.Id property

Description
Identifier for the acetate layer. Write/Read.

Type
String

Example
```
Dim mAcetateLayer
Set mAcetateLayer = Server.CreateObject("aims.AcetateLayer")
mAcetateLayer.Id = "ID:1976"
Response.write mAcetateLayer.Id
```

See Also
AcetateLayer.Name

AcetateLayer.Item method

Description
Returns the acetate layer object by index. Index=1 is the first added object.

Syntax
AcetateLayer.Item(index as Long) as Object

Argument
index Index to reference of object.

Returned Value
Object Acetate object of the index reference.

Example
```
Dim mAcetateLayer
Dim mPointObject
Dim mObject

Set mAcetateLayer = Server.CreateObject("aims.AcetateLayer")
mAcetateLayer.Visible = true
Set mPointObject = Server.CreateObject("aims.PointObject")

mAcetateLayer.Add mPointObject, imsUnit.imsPixel

Set mObject = mAcetateLayer.Item(1)
```

See Also
AcetateLayer.Add
AcetateLayer.Clear
AcetateLayer.Remove

AcetateLayer.Name property

Description
Name of the acetate layer. Write/Read.

Type
String

Example
```
Dim mAcetateLayer
Set mAcetateLayer = Server.CreateObject("aims.AcetateLayer")
mAcetateLayer.Name = "AcetateLayer1"
Response.write mAcetateLayer.Name
```

AcetateLayer.Remove method

Description
Removes the object from the acetate layer by index.

Syntax
AcetateLayer.Remove (index as Long)

Argument
Index Index to reference to the object.

Returned Value
Boolean

Example
```
Dim mAcetateLayer
Dim mPointObject

Set mAcetateLayer = Server.CreateObject("aims.AcetateLayer")
mAcetateLayer.visible = true

Set mPointObject = Server.CreateObject("aims.PointObject")
mPointObject.x = -117.00
mPointObject.y = 45.00

mAcetateLayer.Add mPointObject, 1
mAcetateLayer.Remove 1
```

See Also
AcetateLayer.Add
AcetateLayer.Clear
AcetateLayer.Item

AcetateLayer.Visible property

Description
Visibility of the acetate layer. True indicates that the acetate layer is visible; false indicates it is not visible. Write/Read.

Type
Boolean

Example
```
Dim mAcetateLayer
Set mAcetateLayer = Server.CreateObject("aims.AcetateLayer")
mAcetateLayer.Visible = true
```

AddressMatchInputs.AddressMatch method

Description
Runs a geocoding request.

Syntax
AddressMatchInputs.AddressMatch(MaxCandidates, MinScore)

Arguments
MaxCandidates (as Long)	Maximum number of returned candidates.
MinScore (as Long)	Minimum score of returned candidates.

Returned Value
Boolean

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap connect, "IMSMapService"

Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
if mAddressMatchInputs.MoveFirst() then
   Do while mAddressMatchInputs.MoveNext()
   if mAddressMatchInputs.Id = "STREET" then
      mAddressMatchInputs.Value = "195 S. Center"
   end if
   Loop
end if
mResult = mAddressMatchInputs.AddressMatch(25,60)
```

AddressMatchInputs.AddressMatchResults property

Description
A collection of candidates, or possible locations, of the address being geocoded. Read only.

Type
AddressMatchResults

Example
```
Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
if mAddressMatchInputs.MoveFirst() then
   do
     if mAddressMatchInputs.Id = "STREET" then
       mAddressMatchInputs.Value = "850 BALBOA ST"
     end if
```

```
    if mAddressMatchInputs.Id = "ZONE" then
      mAddressMatchInputs.Value = 94118
    end if
  loop while mAddressMatchInputs.MoveNext()
end if
mResult = mAddressMatchInputs.AddressMatch(5, 80)

Set mAddressMatchResults = mAddressMatchInputs.AddressMatchResults
if mAddressMatchResults.MoveFirst() then
  Set mPointObject = mAddressMatchResults.Point
  Set al = Server.CreateObject("aims.AcetateLayer")
  al.visible = true
  Set arrow = Server.CreateObject("aims.NorthArrowObject")
  arrow.Size=20
  arrow.x = mPointObject.x
  arrow.y = mPointObject.y
  al.Add arrow, 1
  mMap.Layers.Add al
  mMap.Refresh
end if
urlImage     = mMap.GetImageAsUrl()
```

See Also

AddressMatchResults methods and properties

Note

This sample finds address "850 BALBOA ST" with Zip code 94118 in the map, and draws a northarrow on the first candidate (there may be more than one point with this address). The map service for geocoding has to be prepared. In Author, open the AXL file for the map service and set its geooding properties. See "Using ArcIMS" Chapter 3, "Authoring maps" section "Setting geocoding properties" and the section on"Geocoding styles" for more detail. Also refer to "ArcXML Programmer's Reference Guide" tag <GEOCODE> and tags starting with "GC" for more information.

AddressMatchInputs.ClearValues method

Description

Clears the values of all fields.

Syntax

AddressMatchInputs.ClearValues()

Arguments

None

Returned Value

None

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs
Dim mAddressMatchResults

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
mAddressMatchInputs.ClearValues()
```

AddressMatchInputs.Count property

Description

In geocoding, an address is broken down into several parts, depending on the address style used. This Count property is the number of these parts. See the tutorial in this guide for more detailed information on address matching (geocoding) with the ActiveX Connector. Read only.

Type

Long

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs
Dim mAddressMatchResults

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
mCount = mAddressMatchInputs.Count
Response.write CStr(mCount)
```

AddressMatchInputs.Description property

Description
Contains a brief description of the geocoded items. Read only.

Type
String

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs
Dim mAddressMatchResults
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
   if mAddressMatchInputs.MoveFirst() then
      mDesc = mAddressMatchInputs.Description
   end if
```

AddressMatchInputs.Id property

Description
String identifier for the geocoded field. Read only.

Type
String

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
   if mAddressMatchInputs.MoveFirst() then
      id = mAddressMatchInputs.Id
      Response.write id
   end if
```

AddressMatchInputs.InputType property

Description

Contains the type of the geocoded field. Read only.

Type

String

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mAddressMatchInputs = mMap.layers.Item(1).AddressMatchInputs
if mAddressMatchInputs.MoveFirst() then
   mType = mAddressMatchInputs.InputType
   Response.write mType
end if
```

AddressMatchInputs.Label property

Description

Contains a label for the geocoded items. Read only.

Type

String

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.Server = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
   if mAddressMatchInputs.MoveFirst() then
      mLabel = mAddressMatchInputs.Label
      Response.write mLabel
   end if
```

AddressMatchInputs.MoveFirst method

Description
Sets the internal pointer to the first field.

Syntax
AddressMatchInputs.MoveFirst()

Arguments
None

Returned Value
Boolean False indicates list is empty.

Example
```
Dim mArcIMSConnector
Dim mMap

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

if Not mMap.layers.Item(1).AddressMatchInputs is Nothing then
   mMap.Layers.Item(1).AddressMatchInputs.MoveFirst()
end if
```

AddressMatchInputs.MoveLast method

Description
Sets the internal pointer to the last input that the user supplied.

Syntax
AddressMatchInputs.MoveLast()

Arguments
None

Returned Value
Boolean False indicates list is empty.

Example

```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

   if Not mMap.Layers.Item(1).AddressMatchInputs is Nothing then
      mMap.Layers.item(1).AddressMatchInputs.MoveLast()
   end if
```

AddressMatchInputs.MoveNext method

Description

Sets the internal pointer to the next field.

Syntax

AddressMatchInputs.MoveNext()

Arguments

None

Returned Value

Boolean False indicates end of list is reached.

Example

```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

   if Not mMap.Layers.Item(1).AddressMatchInputs is Nothing then
      mMap.Layers.item(1).AddressMatchInputs.MoveNext()
   end if
```

AddressMatchInputs.MovePrevious method

Description
Sets the internal pointer to the previous field.

Syntax
AddressMatchInputs.MovePrevious()

Arguments
None

Returned Value
Boolean False indicates first field is reached.

Example
```
Dim mArcIMSConnector
Dim mMap

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

   if Not mMap.Layers.Item(1).AddressMatchInputs is Nothing then
      mMap.Layers.item(1).AddressMatchInputs.MovePrevious()
   end if
```

AddressMatchInputs.Style property

Description
Contains the name of the geocoding style. Read only.

Type
String

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mStyle

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

   if Not mMap.Layers.Item(1).AddressMatchInputs is Nothing then
      mStyle = mMap.Layers.item(1).AddressMatchInputs.Style

      Response.write mStyle
   end if
```

AddressMatchInputs.Value property

Description
Address value to find. Is used only against street layers that can be geocoded. Write/Read.

Type
String

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
if mAddressMatchInputs.MoveFirst() then
   mAddressMatchInputs.Value = "195 S Center."
end if
```

AddressMatchInputs.Width property

Description
Width of the field. Read only.

Type
Long

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")

mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
   if mAddressMatchInputs.MoveFirst() then
      mWidth = mAddressMatchInputs.Width
      Response.write CStr(mWidth)
   end if
```

AddressMatchResults.Clear method

Description
Clears the results collection of AddressMatchResults after AddressMatchInputs.AddressMatch.

Syntax
AddressMatchResults.Clear

Arguments
None

Returned Value
None

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
if mAddressMatchInputs.MoveFirst() then
   if mAddressMatchInputs.Id = "STREET" then
      mAddressMatchInputs.Value = "195 S Center"
   end if
end if
mAddressMatchInputs.AddressMatch(25,60)
mCount = mAddressMatchInputs.AddressMatchResults.Count

Response.Write CStr(mCount)

AddressMatchResults.Clear()
```

AddressMatchResults.Count property

Description
Number of candidates for the address being geocoded. Read only.

Type
Long

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
```

```
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
if mAddressMatchInputs.MoveFirst() then
   if mAddressMatchInputs.Id = "STREET" then
      mAddressMatchInputs.Value = "195 S Center"
   end if
end if
mAddressMatchInputs.AddressMatch(25,60)
mCount = mAddressMatchInputs.AddressMatchResults.Count

Response.Write CStr(mCount)
```

AddressMatchResults.MoveFirst method

Description

Sets the internal pointer to the first candidate returned for the address being geocoded.

Syntax

AddressMatchResults.MoveFirst()

Arguments

None

Returned Value

Boolean False indicates list is empty.

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
if mAddressMatchInputs.MoveFirst() then
   if mAddressMatchInputs.Id = "STREET" then
      mAddressMatchInputs.Value = "195 S Center"
   end if
end if
mAddressMatchInputs.AddressMatch(25,60)
Set amr = mAddressMatchInputs.AddressMatchResults.Count

amr.MoveFirst.

Response.Write amr.Value
```

AddressMatchResults.MoveLast method

Description
Sets the internal pointer to the last candidate returned for the address being geocoded.

Syntax
AddressMatchResults.MoveLast()

Arguments
None

Returned Value
Boolean

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
if mAddressMatchInputs.MoveFirst() then
   if mAddressMatchInputs.Id = "STREET" then
      mAddressMatchInputs.Value = "195 S Center"
   end if
end if
mAddressMatchInputs.AddressMatch(25,60)
mAddressMatchInputs.AddressMatchResults.MoveLast()
```

AddressMatchResults.MoveNext method

Description
Sets the internal pointer to the next candidate returned for the address being geocoded.

Syntax
AddressMatchResults.MoveNext()

Arguments
None

Returned Value
Boolean

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs
Dim mAddressMatchResults
```

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
if mAddressMatchInputs.MoveFirst() then
   if mAddressMatchInputs.Id = "STREET" then
      mAddressMatchInputs.Value = "195 S Center"
   end if
end if
mAddressMatchInputs.AddressMatch(25,60)
Set mAddressMatchResults = mAddressMatchInputs.AddressMatchResults
mAddressMatchResults.MoveNext()
```

AddressMatchResults.MovePrevious method

Description

Sets the internal pointer to the previous address in the list.

Syntax

AddressMatchResults.MovePrevious()

Arguments

None

Returned Value

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs
Dim mAddressMatchResults
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
if mAddressMatchInputs.MoveFirst() then
   if mAddressMatchInputs.Id = "STREET" then
      mAddressMatchInputs.Value = "195 S Center"
   end if
end if
mAddressMatchInputs.AddressMatch(25,60)
Set mAddressMatchResults = AddressMatchInputs.AddressMatchResults
mAddressMatchResults.MovePrevious
```

AddressMatchResults.Point property

Description
Spatial location of the current candidate. Read only.

Type
PointObject

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs
Dim mAddressMatchResults
Dim mPointObject
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

     Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
     if mAddressMatchInputs.MoveFirst() then
          if mAddressMatchInputs.Id = "STREET" then
               mAddressMatchInputs.Value = "380 New York st"
          end if
     end if
     mResult = mAddressMatchInputs.AddressMatch(25, 60)
     Set mAddressMatchResults = mAddressMatchInputs.AddressMatchResults
     if mAddressMatchResults.MoveFirst() then
          Set mPointObject = mAddressMatchResults.Point
          Set al = Server.CreateObject("aims.AcetateLayer")
          al.visible = true
          Set pt = Server.CreateObject("aims.PointObject")
          pt.x = mPointObject.x
          pt.y = mPointObject.y
          al.Add pt, 1
          mMap.Layers.Add al
          mMap.Refresh
     end if
```

AddressMatchResults.Score property

Description
Relates the degree to which the current candidate matches the address being geocoded. Read only.

Type
Long

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs
Dim mAddressMatchResults
```

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
if Not mMap.Layers.Item(1).AddressMatchInputs is Nothing then
     Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
     if mAddressMatchInputs.MoveFirst() then
          if mAddressMatchInputs.Id = "STREET" then
               mAddressMatchInputs.Value = "380 New York st"
          end if
     end if
     mResult = mAddressMatchInputs.AddressMatch(25,60)
     Set mAddressMatchResults = mAddressMatchInputs.AddressMatchResults
     if mAddressMatchResults.MoveFirst() then
          mScore = mAddressMatchResults.Score
     end if
end if
```

AddressMatchResults.Value property

Description

The address of the current candidate. Read only.

Type

String

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mAddressMatchInputs
Dim mAddressMatchResults
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mResult = mMap.InitMap(mArcIMSConnector, "Redlands")
if Not mMap.Layers.Item(1).AddressMatchInputs is Nothing then
     Set mAddressMatchInputs = mMap.Layers.Item(1).AddressMatchInputs
     if mAddressMatchInputs.MoveFirst() then
          if mAddressMatchInputs.Id = "STREET" then
               mAddressMatchInputs.Value = "380 New York st"
          end if
     end if
     mResult = mAddressMatchInputs.AddressMatch(25,60)
     Set mAddressMatchResults = mAddressMatchInputs.AddressMatchResults
     if mAddressMatchResults.MoveFirst() then
          mValue = mAddressMatchResults.Value
     Response.Write mValue
     end if
end if
```

ArcIMSConnector.ConnectUsing property

Description
Sets a constant for the connection type. TCP=1, HTTP=2. The default is 1. Write/Read.

Type
Long

Example
```
Dim mArcIMSConnector
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ConnectUsing = 1
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
```

See Also
Show_Map.asp

ArcIMSConnector.GetLastError method

Description
Returns the information within the <ERROR> tag in the ArcXML response from last request.

Syntax
ArcIMSConnector.GetLastError()

Arguments
None

Returned Value
String The error's description.

Example
```
Dim mArcIMSConnector
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerPort = 5300
mArcIMSConnector.ServerName = "IMSServer"
. . .
mLastError = mArcIMSConnector.GetLastError()
Response.write mLastError
```

ArcIMSConnector.GetServiceNames method

Description

Returns a collection of the names of the map services running on the site, excluding map services that are stopped.

Syntax

ArcIMSConnector.GetServiceNames()

Arguments

None

Returned Value

Collection Contains names of all Image Server services.

Example

```
Dim mArcIMSConnector
Dim mService
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerPort = 5300
mArcIMSConnector.ServerName = "IMSServer"
Set mService = mArcIMSConnector.GetServiceNames()
For i = 1 to mService.Count
   Response.write mService.Item(i)
Next
```

ArcIMSConnector.Password property

Description

Password for the connection to be used with authenticated services. Write/Read.

Type

String

Example

```
Dim mArcIMSConnector
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ConnectUsing = 2
mArcIMSConnector.ServerURL = "http://www.yoursite.com"
mArcIMSConnector.Username = "myusername"
mArcIMSConnector.Password = "password"
```

See Also

Show_Map_Http.asp

ArcIMSConnector.RequestAXL property

Description

The last ArcXML request sent to the ArcIMS Spatial Server. XML format. See the example on how to display it in the browser. Read only.

Type

String

Example

```
Dim mArcIMSConnector
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "localhost"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "SantaClara"
mMap.Width = 500
mMap.Height = 450
mMap.Refresh
mLastRequest = mArcIMSConnector.RequestAXL
mLastResponse = mArcIMSConnector.ResponseAXL
mLastRequest = FormatXML(mLastRequest)
mLastResponse = FormatXML(mLastResponse)
Response.write "<br>Request:<br>"
Response.Write mLastRequest
Response.Write "<br>Response:<br>"
Response.Write mLastResponse

function FormatXML(str)
   str = Replace(str,">", "&gt;")
   str = Replace(str, "<", "&lt;")
   str = Replace(str, chr(13), "<BR>")

   FormatXML = str
end function
```

ArcIMSConnector.ResponseAXL property

Description

The last ArcXML response sent from the ArcIMS Spatial Server. XML format. Read only.

Type

String

Example

See example for ArcIMSConnector.RequestAXL.

ArcIMSConnector.SendAXLRequest method

Description

Sends the request to the ArcIMS Spatial Server. Note that the ActiveX Connector by default sends requests to the Image Server. Therefore, a GET_FEATURES request must be redirected to the Query Server.

Syntax

ArcIMSConnector.SendAXLRequest(ServiceName, Request) or
ArcIMSConnector.SendAXLRequest(ServiceName&CustomService=Query,Request) when GET_FEATURES

Arguments

Request	ArcXML request.
ServiceName	The name of the requested service.

Returned Value

String	The ArcIMS Spatial Server's response.

Example

```
Dim mArcIMSConnector
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerPort = 5300
mArcIMSConnector.ServerName = "IMSServer"
mRequest = "<?xml version=""1.1""?>"
mRequest = mRequest & "<ARCXML version=""1.1"">"
mRequest = mRequest & "<REQUEST>"
mRequest = mRequest & "<GET_IMAGE>"
mRequest = mRequest & "<PROPERTIES>"
mRequest = mRequest & "<ENVELOPE minx=""" & MinX & """ minY=""" & MinY & """
maxx=""" & MaxX & """ maxy=""" & maxY & """ />"
mRequest = mRequest & "<IMAGESIZE height=""300"" width=""500""/>"
mRequest = mRequest & "<LEGEND font=""Verdana"" width=""650"" titlefontsize=""12""
columns=""4"" cansplit=""true"" splittext=""(continued)"" autoextend=""true""
backgroundcolor=""255,255,255""/>"
mRequest = mRequest  & "<DRAW map=""false""/>"
mRequest = mRequest & "</PROPERTIES>"
mRequest = mRequest & "</GET_IMAGE>"
mRequest = mRequest & "</REQUEST>"
mRequest = mRequest & "</ARCXML>"

mArcIMSConnector.SendAxlRequest("IMSMapService", mRequest)
```

ArcIMSConnector.ServerName property

Description
Contains the ArcIMS Application Server name to connect to. Write/Read.

Type
String

Example
```
Dim mArcIMSConnector
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerPort = 5300
mArcIMSConnector.ServerName = "IMSServer"
```

See Also
Show_Map.asp

ArcIMSConnector.ServerPort property

Description
Contains the ArcIMS Application Server port to connect to. Write/Read.

Type
Long

Example
```
Dim mArcIMSConnector
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "localhost"
mArcIMSConnector.ServerPort = 5300
```

See Also
ArcIMSConnector.ServerName
Show_Map.asp

ArcIMSConnector.ServerUrl property

Description

URL of the site where the ArcIMS application is running. Needed only when the connection type is HTTP: ConnectUsing=2. Write/Read.

Type

String

Example

```
Dim mArcIMSConnector
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ConnectUsing = 2
mArcIMSConnector.ServerURL = "http://www.yoursite.com"
```

See Also

Show_Map_http.asp

ArcIMSConnector.Username property

Description

Contains the username for the connection for use with authenticated services. Write/Read.

Type

String

Example

```
Dim mArcIMSConnector
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ConnectUsing = 2
mArcIMSConnector.ServerURL = "http://www.yoursite.com"
mArcIMSConnector.Username = "myusername"
mArcIMSConnector.Password = "password"
```

See Also

Show_Map_http.asp

CalloutMarkerSymbol.Antialiasing property

Description

Turns antialiasing on/off. Antialiasing allows lines to appear smooth by coloring adjacent pixels in such a way as to give the edge of a line a more gradual fade into the background. Write/Read.

Type

Boolean

Example

```
Dim mCalloutMarkerSymbol
Set mCalloutMarkerSymbol = Server.CreateObject("aims.CalloutMarkerSymbol")
mCalloutMarkerSymbol.Antialiasing = False
```

CalloutMarkerSymbol.BackColor property

Description

Color constant for the background color of a CalloutMarkerSymbol. The default value is imsWhite (16777215). Write/Read.

Type

Long

Example

```
Dim mCalloutMarkerSymbol
Set mCalloutMarkerSymbol = Server.CreateObject("aims.CalloutMarkerSymbol")
mCalloutMarkerSymbol.BackColor = imsColor.imsRed
```

CalloutMarkerSymbol.BoundaryColor property

Description

Color constant for the boundary color of a CalloutMarkerSymbol. The default value is imsBlack (0). Write/Read.

Type

Long

Example

```
Dim mCalloutMarkerSymbol
Set mCalloutMarkerSymbol = Server.CreateObject("aims.CalloutMarkerSymbol")
mCalloutMarkerSymbol.BoundaryColor = imsColor.imsRed
```

CalloutMarkerSymbol.Clone method

Description
Clones the CalloutMarkerSymbol.

Syntax
CalloutMarkerSymbol.Clone()

Arguments
None

Returned Value
CalloutMarkerSymbol Cloned symbol.

Example
```
Dim mClone
Dim mCalloutMarkerSymbol
Set mCalloutMarkerSymbol = Server.CreateObject("aims.CalloutMarkerSymbol")
Set mClone = mCalloutMarkerSymbol.Clone()
```

CalloutMarkerSymbol.Font property

Description
Font name for text labels used with a CalloutMarkerSymbol. The default value is “default”. Write/Read.

Type
String

Example
```
Dim mCalloutMarkerSymbol
Set mCalloutMarkerSymbol = Server.CreateObject("aims.CalloutMarkerSymbol")
mCalloutMarkerSymbol.Font = "Arial"
```

CalloutMarkerSymbol.FontColor property

Description
Font color constant for the font used with CalloutMarkerSymbol. The default value is imsBlack (0). Write/Read.

Type
Long

Example
```
Dim mCalloutMarkerSymbol
Set mCalloutMarkerSymbol = Server.CreateObject("aims.CalloutMarkerSymbol")
mCalloutMarkerSymbol.FontColor = imsColor.imsRed
```

CalloutMarkerSymbol.FontSize property

Description

Font size for the font used with a CalloutMarkerSymbol. The default value is 12. Write/Read.

Type

Long

Example

```
Dim mCalloutMarkerSymbol
Set mCalloutMarkerSymbol = Server.CreateObject("aims.CalloutMarkerSymbol")
mCalloutMarkerSymbol.FontSize = 10
```

CalloutMarkerSymbol.FontStyle property

Description

Font style constant for the font used with a CalloutMarkerSymbol. The default value is imsRegular (0). The possible values are:

0 = imsRegular
1 = imsBold
2 = imsItalic
3 = imsUnderline
4 = imsOutline

Write/Read.

Type

Long

Example

```
Dim mCalloutMarkerSymbol
Set mCalloutMarkerSymbol = Server.CreateObject("aims.CalloutMarkerSymbol")
mCalloutMarkerSymbol.FontStyle = imsFontStyle.imsBold
```

CalloutMarkerSymbol.Glowing property

Description

Glowing color constant for labels used with a CalloutMarkerSymbol. Glowing is the halo effect around text labels. The default value is imsBlack (0). Write/Read.

Type

Long

Example

```
Dim mCalloutMarkerSymbol
Set mCalloutMarkerSymbol = Server.CreateObject("aims.CalloutMarkerSymbol")
mCalloutMarkerSymbol.Glowing = imsColor.imsRed
```

CalloutMarkerSymbol.Interval property

Description

Distance between a point and a callout box. A smaller number will bring the box closer to the point. The default value is 10. Write/Read.

Type

Double

Example

```
Dim mCalloutMarkerSymbol
Set mCalloutMarkerSymbol = Server.CreateObject("aims.CalloutMarkerSymbol")
mCalloutMarkerSymbol.Interval = 5
```

CalloutMarkerSymbol.Outline property

Description

Outline color constant for the CalloutMarkerSymbol. Write/Read.

Type

Long

Example

```
Dim mCalloutMarkerSymbol
Set mCalloutMarkerSymbol = Server.CreateObject("aims.CalloutMarkerSymbol")
mCalloutMarkerSymbol.Outline = imsColor.imsRed
```

CalloutMarkerSymbol.Shadow property

Description

Color constant for the shadow color of a CalloutMarkerSymbol. Write/Read.

Type

Long

Example

```
Dim mCalloutMarkerSymbol
Set mCalloutMarkerSymbol = Server.CreateObject("aims.CalloutMarkerSymbol")
mCalloutMarkerSymbol.Shadow = imsColor.imsRed
```

CalloutMarkerSymbol.Transparency property

Description

Transparency coefficient for a CalloutMarkerSymbol. Smaller numbers indicate more transparency. The default value is 1.0. Write/Read.

Type

Double

Example

```
Dim mCalloutMarkerSymbol
Set mCalloutMarkerSymbol = Server.CreateObject("aims.CalloutMarkerSymbol")
mCalloutMarkerSymbol.Transparency = 1.0
```

Circle.Distance property

Description
Contains the radius of the circle in database units. Write/Read.

Type
Double

Example
```
Dim mCircle
Set mCircle = Server.CreateObject("aims.Circle")
mCircle.Distance = 0.12
```

Circle.X property

Description
Contains the x coordinate of the circle's center. Write/Read.

Type
Double

Example
```
Dim mCircle
Set mCircle = Server.CreateObject("aims.Circle")
mCircle.X = -37.12
```

Circle.Y property

Description
Contains the y coordinate of the circle's center. Write/Read.

Type
Double

Example
```
Dim mCircle
Set mCircle = Server.CreateObject("aims.Circle")
mCircle.Y = 122.67
```

Envelope.Xmax property

Description
Contains the x maximum coordinate. Write/Read.

Type
Double

Example
```
Dim mEnvelope
Set mEnvelope = Server.CreateObject("aims.Envelope")
mEnvelope.Xmax = 12
```

See Also
Make_Spatial_Query.asp
Buffer.asp

Envelope.Xmin property

Description
Contains the x minimum coordinate. Write/Read.

Type
Double

Example
```
Dim mEnvelope
Set mEnvelope = Server.CreateObject("aims.Envelope")
mEnvelope.XMin = 7
```

See Also
Make_Spatial_Query.asp
Buffer.asp

Envelope.Ymax property

Description
Contains the y maximum coordinate. Write/Read.

Type
Double

Example
```
Dim mEnvelope
Set mEnvelope = Server.CreateObject("aims.Envelope")
mEnvelope.YMax = 60
```

See Also
Make_Spatial_Query.asp
Buffer.asp

Envelope.Ymin property

Description
Contains the y minimum coordinate. Write/Read.

Type
Double

Example
```
Dim mEnvelope
Set mEnvelope = Server.CreateObject("aims.Envelope")
mEnvelope.YMin = 13
```

See Also
Make_Spatial_Query.asp
Buffer.asp

FeatureLayer.AddressMatchInputs property

Description

Contains data necessary for geocoding. It is only for geocodable layers. Read only.

Type

AddressMatchInputs

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mAddress

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
if Not mMap.Layers.Item(1).AddressMatchInputs is Nothing then
    Set mAddress = mMap.Layers.Item(1).AddressMatchInputs
end if
```

FeatureLayer.Clone method

Description
Creates the clone of a specific layer.

Syntax
FeatureLayer.Clone()

Arguments
None

Returned Value
FeatureLayer Clone of the feature class layer.

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mClone
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set mClone = mMap.Layers.Item(1).Clone()
mClone.Renderer.Item(1).Symbol.Color = 255
mMap.Layers.Add mClone
mMap.Refresh
```

See Also
Select_and_Highlight.asp
Make_Spatial_Query.asp
Buffer.asp

FeatureLayer.Envelope property

Description
Envelope of a feature layer. Read only.

Type
Envelope

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mClone
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Refresh
Response.Write mMap.Layers.Item(1).Envelope.XMin
Response.Write mMap.Layers.Item(1).Envelope.YMin
Response.Write mMap.Layers.Item(1).Envelope.XMax
Response.Write mMap.Layers.Item(1).Envelope.YMax
```

FeatureLayer.FeatureClass property

Description
Contains the type of a feature class layer—point, line, or polygon. Write/Read.

Type
String

Example
```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
MArcIMSConnector.ServerName = "IMSServer"
MArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mFeatureClass = mMap.Layers.Item(1).FeatureClass
Response.write mFeatureClass
```

FeatureLayer.Filter property

Description

Contains the spatial filter of the layer. FeatureLayer filter affects only the output image of the layer. Write/Read.

Type

Filter

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mFilter
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "World"
Set mFilter = mMap.Layers.Item(1).Filter
```

See Also

Buffer.asp
Filter properties and methods
Make_Spatial_Query.asp
Recordset.Filter
Select_and_Highlight.asp

FeatureLayer.Id property

Description

String identifier for the layer. Write/Read.

Type

String

Example

```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mId = mMap.Layers.Item(1).Id
Response.write mId
```

FeatureLayer.MaxScale property

Description

The maximum scale in map units per pixel that you want the ActiveX Connector to use to display the layer. If the layer's Maxscale attribute value is defined in the <LAYERINFO> tag of the configuration file, the ActiveX Connector uses that value. If the connector returns a value of 0, this signifies that no value is defined in the layer's <LAYERINFO> tag's Maxscale attribute. Read only.

See the *ArcXML Programmer's Reference Guide* <LAYERINFO> tag for information on how to convert between a relative scale to map units per pixel.

Type

Double

See Also

FeatureLayer.MinScale

FeatureLayer.MinScale property

Description

The minimum scale in map units per pixel that you want the ActiveX Connector to use to display the layer. If the layer's Minscale attribute value is defined in the <LAYERINFO> tag of the configuration file, the ActiveX Connector uses that value. If the connector returns a value of 0, this signifies that no value is defined in the layer's <LAYERINFO> tag's Maxscale attribute. Read only.

See *ArcXML Programmer's Reference Guide* <LAYERINFO> tag for information on how to convert between a relative scale to map units per pixel.

Type

Double

See also

FeatureLayer.MaxScale

FeatureLayer.Name property

Description

Name of the layer. Write/Read.

Type

String

Example

```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mName = mMap.Layers.Item(1).Name
Response.write mName
```

FeatureLayer.Recordset property

Description

Contains the layer's Recordset object. Recordset is valid only for layers that have a specified ID. Read only.

Type

Recordset

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mRecordset
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set env = Server.CreateObject("aims.Envelope")
env.Xmax = 25
env.Xmin = -25
env.Ymax = 25
env.Ymin = -25
mMap.Layers.Item(1).Recordset.Envelope = env
mMap.Layers.Item(1).Recordset.MoveFirst
For i = 1 to mMap.Layers.Item(1).Recordset.Fields.Count
Response.write mMap.Layers.Item(1).Recordset.Fields.FieldValueAsString(i)
Next
```

See Also

Identify.asp
Query_Attribute_Data.asp
Recordset properties and methods

FeatureLayer.Renderer property

Description

Layer's renderer as a Renderer object. Write/Read.

Type

Object

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mRenderer
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Response.Write TypeName(mMap.Layers.Item(1).Renderer)
mMap.Refresh
```

See Also

Buffer.asp
Change_Marker_Symbol.asp
Change_Polygon_Symbol.asp
GroupRenderer properties and methods
Make_Spatial_Query.asp
ScaleDependentRenderer properties and methods
Select_and_Highlight.asp
SimpleLabelRenderer properties and methods
SimpleRenderer properties and methods
ValueMapLabelRenderer properties and methods
ValueMapRenderer properties and methods

FeatureLayer.Visible property

Description
Visibility of the layer. Write/Read.

Type
Boolean

Example
```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Layers.Item(1).Visible = True
mMap.Refresh
```

See Also
Change_layer_visibility.asp
Change_Marker_Symbol.asp
Change_Polygon_Symbol.asp
Make_Spatial_Query.asp

FeatureLayer.Workspace property

Description

Exists only for a layer created through the method Layers.Create(); otherwise, this property is Nothing. There are two types of workspace objects for featureclass layers: ShapeWorkspace and SDEWorkspace. Read only.

Type

Object

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mShapeWorkspace
Dim mLayer
Dim mWorkspace
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
Set mMap = Server.CreateObject("aims.Map")
Set mShapeWorkspace = Server.CreateObject("aims.ShapeWorkspace")
mShapeWorkspace.Name = "Roads"
mShapeWorkspace.Directory = "D:\EsriData\World"
mShapeWorkspace.FeatureClass = imsFeatureClass.imsLine
mMap.InitMap mArcIMSConnector, "World"
Set mLayer = mMap.Layers.Create(mShapeWorkspace)
mMap.Layers.Add(mLayer)
mMap.Refresh
Response.write mLayer.Workspace.Name
```

See Also

imsFeatureClass
Layers.Create
SDEWorkspace properties and methods
ShapeWorkspace properties and methods

Fields.Count property

Description
Number of fields. Read only.

Type
Long

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mFields
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap  mArcIMSConnector, "IMSMapService"
if mMap.Layers.Item(1).Recordset.movefirst then
   Set mFields = mMap.Layers.Item(1).Recordset.Fields
   mCount = mFields.Count
   Response.Write CStr(mCount)
end if
```

Fields.FieldValue method

Description
Returns the variant value of the field by index.

Syntax
Fields.FieldValue(index as Long)

Argument
Index as long Index to reference the value.

Returned Value
Variant Value of the field.

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mFields
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set mFields = mMap.Layers.Item(1).Recordset.Fields
mValue = mFields.FieldValue(1)
if mMapLayers.Item(1).Recordset.Movefirst then
   set mFields = mMap.Layers.Item(1).Recordset.Fields
   mValue = mFields.FieldValue(1)
   Response.write mValue
end if
```

See Also

Fields.FieldValueAsString
Query_Attribute_Data.asp

Fields.FieldValueAsString method

Description

Returns the string value of the field by index.

Syntax

Fields.FieldValueAsString(index as Long)

Argument

Index as long Index to reference the value.

Returned Value

String String value of the field.

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mFields
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
if mMapLayers.Item(1).Recordset.Movefirst then
   set mFields = mMap.Layers.Item(1).Recordset.Fields
   mValue = mFields.FieldValueAsString(1)
   Response.write mValue
end if
Set mFields = mMap.Layers.Item(1).Recordset.Fields
mValue = mFields.FieldValueAsString(1)
Response.write mValue
```

See Also

Fields.FieldValue
Identify.asp

Filter.AddGObject method

Description

Adds the object to the spatial filter collection.

A filter allows only one spatial object per spatial query; therefore, this method should be called only once per query. If a query needs more than one spatial object, such as two envelopes, you can create a polygon that contains the two envelopes as rings.

Syntax

Filter.AddGObject(vNewValue as Object)

Argument

GObject as Object Spatial filter object.

Returned Value

None

Example

```
Dim mArcIMSConnector
Dim mMap
Dim spobj
Dim origLayer
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSService"
Set spobj =  Server.CreateObject("aims.Envelope")
spobj.Xmin = -122.4365
spobj.Ymin = 37.784
spobj.Xmax = -122.433
spobj.Ymax = 37.785
Set origLayer = mMap.Layers.Item(2).Clone()
mMap.Layers.Add origLayer
origLayer.Name = "Filter"
origLayer.Filter.AddGObject spObj
mMap.mLayers.Add origLayer
mMap.Refresh
```

See Also

Buffer.asp
Filter.ClearGObjects
Filter.Count
Filter.GObject
Filter.RemoveGObject
Identify.asp
Make_Spatial_Query.asp
Select_and_Highlight.asp

Filter.AddSubField method

Description

Sets the list of fields returned after querying. Multiple subfields can be added to the filter object.
If subfields are not used, all fields are returned. If subfields are used, only the added fields are returned. The subfields list can include fields from the layer table or a joined table.

Syntax

Filter.AddSubField(field name as String)

Argument

FieldName as string

Returned Value

None

Example

```
Dim mFilter
Dim mPointObject
Set mFilter = Server.CreateObject("aims.Filter")
Set mPointObject = Server.CreateObject("aims.PointObject")
mFilter.AddGObject(mPointObject)
mFilter.addsubfield("AREA")
mFilter.addsubfield("BKG_KEY")
mFilter.addsubfield("POP1990")
mFilter.addsubfield("POP90_SQMI")
mFilter.addsubfield("HOUSEHOLDS")
```

Filter.Buffer property

Description

Buffer area width, in the units defined by Filter.BufferUnits. Write/Read.

Type

Double

Example

```
Dim mArcIMSConnector
Dim mMap
Dim spobj
Dim origLayer
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSService"
Set spobj = Server.CreateObject("aims.Envelope")
spobj.Xmin = -122.4365
spobj.Ymin = 37.784
spobj.Xmax = -122.433
spobj.Ymax = 37.785
Set origLayer = mMap.Layers.Item(2).Clone()
origLayer.Renderer.Symbol.Color = 255
origLayer.Name = "Filter"
```

```
origLayer.Filter.AddGObject spObj
origLayer.Filter.BufferUnits = 1
origLayer.Filter.Buffer = 10
Set lay = mMap.Layers.CreateBuffer(origLayer.Filter, origLayer)
mMap.mLayers.Add lay
mMap.Refresh
```

See Also

Buffer.asp
Filter.BufferUnits

Filter.BufferUnits property

Description

Specifies the units that apply to the buffer. Use imsMapUnits constant. Write/Read.

Type

Long

Example

```
Dim mArcIMSConnector
Dim mMap
Dim spobj
Dim origLayer
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSService"
Set spobj = Server.CreateObject("aims.Envelope")
spobj.Xmin = -122.4365
spobj.Ymin = 37.784
spobj.Xmax = -122.433
spobj.Ymax = 37.785
Set origLayer = mMap.Layers.Item(2).Clone()
origLayer.Renderer.Symbol.Color = 255
origLayer.Name = "Filter"
origLayer.Filter.AddGObject spObj
origLayer.Filter.BufferUnits = 1
origLayer.Filter.Buffer = 10
Set lay = mMap.Layers.CreateBuffer(origLayer.Filter, origLayer)
mMap.mLayers.Add lay
mMap.Refresh
```

See Also

imsMapUnits

Filter.ClearGObjects method

Description
Removes all objects from the spatial filter collection.

Syntax
Filter.ClearGObjects()

Arguments
None

Returned Value
None

Example
```
Dim mFilter
Dim mPointObject
Set mFilter = Server.CreateObject("aims.Filter")
Set mPointObject = Server.CreateObject("aims.PointObject")
mFilter.AddGObject(mPointObject)
mFilter.ClearGObjects()
```

See Also
Filter.AddGObject
Filter.RemoveGObject
Identify.asp
Select_and_Highlight.asp

Filter.ClearSubField method

Description
Deletes the subfields added to the filter object.

Syntax
Filter.ClearSubFields()

Arguments
None

Returned Value
None

Example
```
Dim mFilter
Dim mPointObject
Set mFilter = Server.CreateObject("aims.Filter")
Set mPointObject = Server.CreateObject("aims.PointObject")
mFilter.AddGObject(mPointObject)
mFilter.addsubfield("AREA")
mFilter.addsubfield("BKG_KEY")
mFilter.addsubfield("POP1990")
mFilter.addsubfield("POP90_SQMI")
mFilter.addsubfield("HOUSEHOLDS")
For x = 1 to mFilter.GetSubFieldsCount
Response.Write "<BR>" & mFilter.Getsubfield(CInt(i))
Next x
mFilter.ClearSubfields
```

Filter.Count property

Description
Number of spatial filter objects. Read only.

Type
Long

Example
```
Dim mFilter
Dim mPointObject
Set mFilter = Server.CreateObject("aims.Filter")
Set mPointObject = Server.CreateObject("aims.PointObject")
mPointObject.x = -125.0
mPointObject.y = 45.0

mFilter.AddGObject mPointObject
mCount = mFilter.Count
Response.write CStr(mCount)
```

See Also
Filter.AddGObject
Filter.ClearGObjects
Filter.RemoveGObject

Filter.GetSubField method

Description
Returns the list of subfields added to the filter object.

Syntax
Filter.GetSubField

Argument
Index as long

Returned Value
Subfield name as string

Example
```
Dim mFilter
Dim mPointObject
Set mFilter = Server.CreateObject("aims.Filter")
Set mPointObject = Server.CreateObject("aims.PointObject")
mFilter.AddGObject(mPointObject)
mFilter.addsubfield("AREA")
mFilter.addsubfield("BKG_KEY")
mFilter.addsubfield("POP1990")
mFilter.addsubfield("POP90_SQMI")
mFilter.addsubfield("HOUSEHOLDS")
For x = 1 to mFilter.GetSubFieldsCount
Response.Write "<BR>" & mFilter.Getsubfield(CInt(i))
Next x
```

Filter.GetSubFieldsCount method

Description
Returns the count of subfields added to the filter object.

Syntax
Filter.GetSubFieldsCount()

Arguments
None

Returned Value
Subfields count as long

Example
```
Dim mFilter
Dim mPointObject
Set mFilter = Server.CreateObject("aims.Filter")
Set mPointObject = Server.CreateObject("aims.PointObject")
mFilter.AddGObject(mPointObject)
mFilter.addsubfield("AREA")
mFilter.addsubfield("BKG_KEY")
mFilter.addsubfield("POP1990")
mFilter.addsubfield("POP90_SQMI")
mFilter.addsubfield("HOUSEHOLDS")
For x = 1 to mFilter.GetSubFieldsCount
Response.Write "<BR>" & mFilter.Getsubfield(CInt(i))
Next x
```

Filter.GObject method

Description
Returns the object from the spatial filter collection by index.

Syntax
Object Filter.GObject(index as Long)

Argument
Index Index for reference to the object.

Returned Value
Object Spatial filter object.

Example
```
Dim mFilter
Dim mPointObject
Dim mGObject
Set mFilter = Server.CreateObject("aims.Filter")
Set mPointObject = Server.CreateObject("aims.PointObject")
mFilter.AddGObject mPointObject
Set mGObject = mFilter.GObject(1)
```

See Also
Filter.AddGObject
Filter.ClearGObjects
Filter.RemoveGObject

Filter.JoinTables property

Description
Contains a list of join tables separated by blank spaces (only for an ArcSDE layer). Write/Read.

Type
String

Example
```
index=mMap.Layers.Find("sdelayerID")
set record=mMap.Layers.item(index).Recordset
record.Filter.JoinTables="database.anothertable"
record.Filter.WhereExpression="database.sdelayerName.STATE_NAME='Alabama' and
database.anothertable.STATE_NAME=database.sdelayerName.STATE_NAME"
```

See Also
Filter.WhereExpression

Filter.RemoveGObject method

Description
Removes the object from the spatial filter collection by index.

Syntax
Filter.RemoveGObject(index as Long)

Argument
Index as long Index for reference to object.

Returned Value
Boolean

Example
```
Dim mFilter
Dim mPointObject
Set mFilter = Server.CreateObject("aims.Filter")
Set mPointObject = Server.CreateObject("aims.PointObject")
mFilter.AddGObject(mPointObject)
mResult = mFilter.RemoveGObject(1)
```

See Also
Filter.AddGObject
Filter.ClearGObjects
Filter.Count

Filter.RemoveSubField method

Description
Deletes a subfield added to the filter object.

Syntax
Filter.RemoveSubField

Argument
Index as long

Returned Value
None

Example
```
Dim mFilter
Dim mPointObject
Set mFilter = Server.CreateObject("aims.Filter")
Set mPointObject = Server.CreateObject("aims.PointObject")
mFilter.AddGObject(mPointObject)
mFilter.addsubfield("AREA")
mFilter.addsubfield("BKG_KEY")
mFilter.addsubfield("POP1990")
mFilter.addsubfield("POP90_SQMI")
mFilter.addsubfield("HOUSEHOLDS")
For x = 1 to mFilter.GetSubFieldsCount
Response.Write "<BR>" & mFilter.Getsubfield(CInt(i))
Next x
For x = 1 to mFilter.GetSubFielsCount
mFilter.RemoveSubfield(1)
Next x
```

Filter.WhereExpression property

Description
The SQL where clause, the standard used for querying features. Write/Read.

Type
String

Example
```
Dim mFilter
Set mFilter = Server.CreateObject("aims.Filter")
mFilter.WhereExpression = "#ID# < 1976"
```

See Also
Query_Attribute_Data.asp

GradientFillSymbol.Antialiasing property

Description

Boolean value indicating whether or not antialiasing will be used while rendering. Antialiasing allows lines to appear smooth by coloring adjacent pixels in such a way as to give the edge of a line a more gradual fade into the background. Write/Read.

Type

Boolean

Example

```
Dim mGradientFillSymbol
Set mGradientFillSymbol = Server.CreateObject("aims.GradientFillSymbol")
mGradientFillSymbol.Antialiasing = True
```

GradientFillSymbol.Boundary property

Description

Boolean value indicating whether or not a boundary will be rendered for a GradientFillSymbol. Write/Read.

Type

Boolean

Example

```
dim mGradientFillSymbol
set mGradientFillSymbol = Server.CreateObject("aims.GradientFillSymbol")
mGradientFillSymbol.Boundary = True
```

GradientFillSymbol.Clone method

Description
Clones the GradientFillSymbol.

Syntax
GradientFillSymbol.Clone()

Arguments
None

Returned Value
GradientFillSymbol Cloned symbol.

Example
```
dim mClone
dim mGradientFillSymbol
Set mGradientFillSymbol = Server.CreateObject("aims.GradientFillSymbol")
Set mClone = mGradientFillSymbol.Clone()
```

GradientFillSymbol.FinishColor property

Description
Ending color constant for a GradientFillSymbol. The default value is imsGreen (65280). Write/Read.

Type
Long

Example
```
Dim mGradientFillSymbol
Set mGradientFillSymbol = Server.CreateObject("aims.GradientFillSymbol")
mGradientFillSymbol.FinishColor = imsColor.imsBlue
```

GradientFillSymbol.GradientStyle property

Description

Constant indicating the fill style of the GradientFillSymbol. The default value is imsBDiagonal (0). The possible values are:

0 = imsBDiagonal
1 = imsFDiagonal
2 = imsHorizontal
3 = imsVertical

Write/Read.

Type

Long

Example

```
Dim mGradientFillSymbol
Set mGradientFillSymbol = Server.CreateObject("aims.GradientFillSymbol")
mGradientFillSymbol.GradientStyle = imsGradientStyle.imsVertical
```

See Also

imsGradientStyle

GradientFillSymbol.Overlap property

Description

Boolean value indicating whether or not labels will be allowed to overlap a GradientFillSymbol. Write/Read.

Type

Boolean

Example

```
Dim mGradientFillSymbol
Set mGradientFillSymbol = Server.CreateObject("aims.GradientFillSymbol")
mGradientFillSymbol.Overlap = True
```

GradientFillSymbol.StartColor property

Description

Color constant for the start color of a GradientFillSymbol. The default value is imsRed (255). Write/Read.

Type

Long

Example

```
Dim mGradientFillSymbol
Set mGradientFillSymbol = Server.CreateObject("aims.GradientFillSymbol")
mGradientFillSymbol.StartColor = imsColor.imsRed
```

GradientFillSymbol.Transparency property

Description

Transparency coefficient for a GradientFillSymbol. Smaller numbers indicate more transparency. The default value is 1.0. Write/Read.

Type

Double

Example

```
Dim mGradientFillSymbol
Set mGradientFillSymbol = Server.CreateObject("aims.GradientFillSymbol")
mGradientFillSymbol.Transparency = 1.0
```

GroupRenderer.Add method

Description
Adds a renderer to the GroupRenderer object.

Syntax
GroupRenderer.Add(Renderer as Object)

Argument
Renderer Renderer object to add.

Returned Value
Boolean

Example
```
Dim mGroupRenderer
Dim mRenderer
Set mGroupRenderer = Server.CreateObject("aims.GroupRenderer")
Set mRenderer = Server.CreateObject("aims.SimpleRenderer")
mResult = mGroupRenderer.Add(mRenderer)
```

GroupRenderer.Clear method

Description
Removes all renderers from the GroupRenderer.

Syntax
GroupRenderer.Clear()

Arguments
None

Returned Value
None

Example
```
Dim mGroupRenderer
Dim mRenderer
Set mGroupRenderer = Server.CreateObject("aims.GroupRenderer")
Set mRenderer = Server.CreateObject("aims.SimpleRenderer")
mResult = mGroupRenderer.Add(mRenderer)
mGroupRenderer.Clear()
```

GroupRenderer.Clone method

Description
Clones the GroupRenderer.

Syntax
GroupRenderer.Clone()

Arguments
None

Returned Value
GroupRenderer Cloned renderer.

Example
```
Dim mClone
Dim mGroupRenderer
Set mGroupRenderer = Server.CreateObject("aims.GroupRenderer")
Set mClone = mGroupRenderer.Clone()
```

GroupRenderer.Count property

Description
Number of renderers in the GroupRenderer. Read only.

Type
Long

Example
```
Dim mGroupRenderer
Dim mRenderer
Set mGroupRenderer = Server.CreateObject("aims.GroupRenderer")
Set mRenderer = Server.CreateObject("aims.SimpleRenderer")
mResult = mGroupRenderer.Add(mRenderer)
mCount = mGroupRenderer.Count
```

GroupRenderer.Item method

Description
Returns the renderer in the GroupRenderer at the specified index.

Syntax
GroupRenderer.Item(index as Long)

Argument
Index as long Indicates the position of the renderer in the GroupRenderer collection.

Returned Value
Object Any renderer.

Example
```
Dim mGroupRenderer
Dim mRenderer
Set mGroupRenderer = Server.CreateObject("aims.GroupRenderer")
Set mRenderer = Server.CreateObject("aims.SimpleRenderer")
mResult = mGroupRenderer.Add(mRenderer)
Set mRenderer = mGroupRenderer.Item(index)
```

See Also
Buffer.asp

GroupRenderer.Remove method

Description
Removes a renderer from a GroupRenderer by index.

Syntax
GroupRenderer.Remove(index as Long)

Argument
Index Index to reference to the renderer.

Returned Value
Boolean

Example
```
Dim mGroupRenderer
Dim mRenderer
Set mGroupRenderer = Server.CreateObject("aims.GroupRenderer")
Set mRenderer = Server.CreateObject("aims.SimpleRenderer")
mResult = mGroupRenderer.Add(mSimpleRenderer)
mResult = mGroupRenderer.Remove(1)
```

HashLineSymbol.Antialiasing property

Description

Boolean value indicating whether or not antialiasing will be used while rendering a HashLineSymbol. Antialiasing allows lines to appear smooth by coloring adjacent pixels in such a way as to give the edge of a line a more gradual fade into the background. Write/Read.

Type

Boolean

Example

```
Dim mHashLineSymbol
Set mHashLineSymbol = Server.CreateObject("aims.HashLineSymbol")
mHashLineSymbol.Antialiasing = True
```

HashLineSymbol.Clone method

Description

Clones the HashLineSymbol.

Syntax

HashLineSymbol.Clone()

Arguments

None

Returned Value

HashLineSymbol Cloned symbol.

Example

```
Dim mClone
Dim mHashLineSymbol
Set mHashLineSymbol = Server.CreateObject("aims.HashLineSymbol")
Set mClone = mHashLineSymbol.Clone()
```

HashLineSymbol.Color property

Description

Color constant for the HashLineSymbol. The default value is imsBlack (0). Write/Read.

Type

Long

Example

```
Dim mHashLineSymbol
Set mHashLineSymbol = Server.CreateObject("aims.HashLineSymbol")
mHashLineSymbol.Color = imsColor.imsRed
```

See Also

imsColor

HashLineSymbol.Interval property

Description

Distance between railroad crosshatches on the HashLineSymbol. The distance units are pixels. The default value is 8. Write/Read.

Type

Long

Example

```
Dim mHashLineSymbol
Set mHashLineSymbol = Server.CreateObject("aims.HashLineSymbol")
mHashLineSymbol.Interval = 10
```

HashLineSymbol.LineThickness property

Description

Line thickness in pixels of the HashLineSymbol. The default value is 1. Write/Read. See *ArcXML Programmer's Reference Guide* <HASHLINESYMBOL> tag for the different components that make up HASHLINESYMBOL.

Type

Long

Example

```
Dim mHashLineSymbol
Set mHashLineSymbol = Server.CreateObject("aims.HashLineSymbol")
mHashLineSymbol.LineThickness = 3
```

HashLineSymbol.Overlap property

Description

Boolean value indicating whether or not labels will overlap a HashLineSymbol. The default value is True. Write/Read.

Type

Boolean

Example

```
Dim mHashLineSymbol
Set mHashLineSymbol = Server.CreateObject("aims.HashLineSymbol")
mHashLineSymbol.Overlap = False
```

HashLineSymbol.Style property

Description

Style constant that specifies a line style for a HashLineSymbol. The default value is imsForeGround (0). The possible values are:

0 = imsForeGround
1 = imsBackGround

Write/Read.

Type

Long

Example

```
Dim mHashLineSymbol
Set mHashLineSymbol = Server.CreateObject("aims.HashLineSymbol")
mHashLineSymbol.Style = imsGroundStyle.imsBackGround
```

See Also

imsGroundStyle

HashLineSymbol.TickThickness property

Description

Tick thickness in pixels for the HashLineSymbol. The default value is 1. Write/Read. See *ArcXML Programmer's Reference Guide* <HASHLINESYMBOL> tag for the different components that make up HASHLINESYMBOL.

Type

Long

Example

```
Dim mHashLineSymbol
Set mHashLineSymbol = Server.CreateObject("aims.HashLineSymbol")
mHashLineSymbol.TickThickness = 2
```

HashLineSymbol.Transparency property

Description

Transparency coefficient for the HashLineSymbol. Lower numbers will display the symbol with greater transparency. The default value is 1.0. Write/Read.

Type

Double

Example

```
Dim mHashLineSymbol
Set mHashLineSymbol = Server.CreateObject("aims.HashLineSymbol")
mHashLineSymbol.Transparency = 1.0
```

HashLineSymbol.Width property

Description

Symbol width in pixels of a HashLineSymbol. The default value is 6. Write/Read. See *ArcXML Programmer's Reference Guide* <HASHLINESYMBOL> tag for the different components that make up HASHLINESYMBOL.

Type

Double

Example

```
Dim mHashLineSymbol
Set mHashLineSymbol = Server.CreateObject("aims.HashLineSymbol")
mHashLineSymbol.Width = 10
```

ImageLayer.Id property

Description
String identifier for the image layer. Write/Read.

Type
String

Example
```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mId = mMap.Layers.Item(1).Id
Response.write mId
```

ImageLayer.MaxScale property

Description
Contains the maximum scale at which to display the layer. Read only.

Type
Double

Example
```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Response.Write mMap.Layers.Item(1).MaxScale
mMap.Refresh
```

See Also
ImageLayer.MinScale

ImageLayer.MinScale property

Description
Contains the minimum scale at which to display the layer. Read only.

Type
Double

Example
```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Response.Write mMapLayers.Item(1).MinScale
mMap.Refresh
```

See Also
ImageLayer.MaxScale

ImageLayer.Name property

Description
Name of the image layer. Write/Read.

Type
String

Example
```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mName = mMap.Layers.Item(1).Name
Response.write mName
```

ImageLayer.Visible property

Description
Visibility of the layer. Write/Read.

Type
Boolean

Example
```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Layers.Item(1).Visible = True
mMap.Refresh
```

ImageLayer.Workspace property

Description
Exists only for the layer created through method Layers.Create(). It contains the path to the data. Workspace object for image layer. Read only.

Type
Object

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mImageWorkspace
Dim mLayer
Dim mWorkspace

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
Set mImageWorkspace = Server.CreateObject("aims.ImageWorkspace")

mImageWorkspace.Name = "World.tif"
mImageWorkspace.Directory = "D:\EsriData\World"
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set mLayer = mMap.Layers.Create(mImageWorkspace)
mMap.Layers.Add  mLayer
Set mWorkspace = mLayer.Workspace
mMap.Refresh
```

See Also
ImageWorkspace properties and methods
Layers.Create

ImageWorkspace.Directory property

Description

Contains a directory containing images. Write/Read

Type

String

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mImageWorkspace
Dim mLayer

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mImageWorkspace = Server.CreateObject("aims.ImageWorkspace")
mImageWorkspace.Directory = "C:\Esridata\Images"
mImageWorkspace.Name = "World.tif"

Set mLayer = MMap.Create(mImageWorkspace)
mMap.Layers.Add mLayer
mMap.Refresh
```

ImageWorkspace.Id property

Description

String identifier for the Image Workspace; Write/Read

Type

String

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mLayer
Dim mImageWorkspace

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mImageWorkspace = Server.CreateObject("aims.ImageWorkspace")
mImageWorkspace.Id = "ImageWorkspace:1"
mImageWorkspace.Directory = "C:/world/tiffs"
mImageWorkspace.Name = "world.tiff"

Set mLayer = mMap.Layers.Create(mImageWorkspace)
mMap.Layers.Add mLayer
mMap.Refresh
```

ImageWorkspace.Name property

Description

Name of the file. Write/Read.

Type

String

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mLayer
Dim mImageWorkspace

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mImageWorkspace = Server.CreateObject("aims.ImageWorkspace")
mImageWorkspace.Id = "ImageWorkspace:1"
mImageWorkspace.Directory = "C:/world/tiffs"
mImageWorkspace.Name = "world.tiff"

Set mLayer = mMap.Layers.Create(mImageWorkspace)
mMap.Layers.Add mLayer
mMap.Refresh
```

imsArrowType constants

Description

Defines the type of North arrow.

Constants

imsType1	1	
imsType2	2	
imsType3	3	
imsType4	4	
imsType5	5	
imsType6	6	
imsType7	7	
imsType8	8	

Returned Value

Long Type of North arrow

Example

```
Dim mArcIMSConnector
Dim mMap
Dim al
Dim arrow
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.width = 500
mMap.Height = 450
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = true
Set arrow = Server.CreateObject("aims.NorthArrowObject")
arrow.x = 25
arrow.y = 25

arrow.arrowtype = 1
arrow.size = 25
al.add arrow, 0
mMap.Layers.Add al
mMap.Refresh
```

imsCapStyle constants

Description

Contains line end or cap style.

Constants

imsButt	1	Butted type.
imsRound	0	Round type.
imsSquare	2	Square type.

Returned Value

Long Line and style.

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
mMap.Layers.Item(1).Renderer.Symbol.CapStyle = 1
mMap.Refresh
```

imsColor constants

Description

Contains color constants.

Constants	Values
imsBlack	0
imsRed	255
imsGreen	65280
imsBlue	16711680
imsMagenta	16711935
imsCyan	16776960
imsWhite	16777215
imsLightgray	13882323
imsDarkgray	11119017
imsGray	8421504
imsLightyellow	14745599
imsYellow	65535
imsLimegreen	3329330
imsTeal	8421376
imsDarkgreen	25600
imsMaroon	128
imsPurple	8388736
imsOrange	42495
imsKhaki	9234160
imsOlive	32896
imsBrown	2763429
imsNavy	8388608

Returned Value

Long Color constants.

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
mMap.Layers.Item(1).Renderer.Symbol.Color = 255
mMap.Refresh
```

imsDirection constants

Description

Defines the pan direction.

Constants

imsNone	0	No direction.
imsNorth	1	North direction.
imsNorthEast	2	Northeast direction.
imsEast	3	East direction.
imsSouthEast	4	Southeast direction.
imsSouth	5	South direction.
imsSouthWest	6	Southwest direction.
imsWest	7	West direction.
imsNorthWest	8	Northwest direction.

Returned Value

Long Defines direction.

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
mMap.Refresh

If Action = "NorthEast" Then
mMap.DoPan 2, 3
mMap.Refresh
end if
```

imsFeatureClass constants

Description

Contains the type of featureclass layer.

Constants

imsPoint	1	Point.
imsLine	2	Line.
imsPolygon	3	Polygon.

Returned Value

Long Featureclass layer type.

Example

```
imsFeatureClass.imsLine
```

imsFieldType constants (field type descriptions)

Description

FieldType constants are not the enumeration type as other constants are. They are descriptions of field types that correspond to number values. Details for these values are in the <FIELD> tag in the *ArcXML Programmer's Reference Guide.*

Field descriptions

-99	Row_id fields
-98	Shape fields
-7	Boolean fields (true|false)
-5	Big integer fields
1	CHAR fields
4	Integer fields.
5	Small integer fields
6	Float fields
8	Double fields
12	String fields of any length

Returned Value

Integer

imsEquality constants

Description

Determines whether or not the bounds of the range (the bounding values) are included in a set or returned range, as follows:

all—all values in the range are returned or set

lower—all values excluding the highest value in the range are returned or set

upper—all values excluding the lowest value in the range are returned or set

Constants

imsAll	0
imsUpper	1
imsLower	2
imsNone	3

Returned Value

Long

Example

```
Dim mValueMapRenderer
Dim mSimpleMarkerSymbol
Set mValueMapRenderer = Server.CreateObject ("aims.ValueMapRenderer")
mValueMapRenderer.LookupFields = "COUNT"
Set mSimpleMarkerSymbol = Server.Create.Object ("aims.SimpleMarkerSymbol")
mResult = mValueMapRenderer.Add(mSimpleMarkerSymbol.imsRange.ims.Range,0,10000.
imsMethod.imsAll,0)
mMethod = mValueMapRenderer.Equality(1)
```

imsFillStyle constants

Description

Contains the fill type.

Constants

imsSolid	0	Solid fill type.
imsBDiagonal	1	Backward diagonal fill type.
imsFDiagonal	2	Forward diagonal fill type.
imsCross	3	Cross fill type.
imsDiagcross	4	Diagonal cross fill type.
imsHorizontal	5	Horizontal fill type.
imsVertical	6	Vertical fill type.
imsLightgray	7	Light gray fill type.
imsGray	8	Gray fill type.
imsDarkgray	9	Dark gray fill type.

Returned Value

Long

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
mMap.Layers.Item(1).Renderer.Item(1).Symbol.FillStyle = 1
mMap.Refresh
```

imsFontStyle constants

Description

Contains the font style.

Constants

imsRegular	0	Regular font style.
imsBold	1	Bold font style.
imsItalic	2	Italic font style.
imsUnderline	3	Underline font style.
imsOutline	4	Outline font style.

Returned Value

Long

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
mMap.Layers.Item(1).Renderer.Item(2).Symbol.FontStyle = 1
mMap.Refresh
```

imsGlobalColor constants

Description

Contains color constants.

Constants	Values
imsAliceblue	16775408
imsAntiquewhite	14150650
imsAqua	16776960
imsAquamarine	13959039
imsAzure	16777200
imsBeige	14480885
imsBisque	12903679
imsBlack	0
imsBlanchedalmond	13495295
imsBlue	16711680
imsBlueviolet	14822282
imsBrown	2763429
imsBurlywood	8894686
imsCadetblue	10526303
imsChartreuse	65407
imsChocolate	1993170
imsCoral	5275647
imsCornflowerblue	15570276
imsCornsilk	14481663
imsCrimson	3937500
imsCyan	16776960
imsDarkblue	9109504
imsDarkcyan	9145088
imsDarkgoldenrod	755384
imsDarkgray	11119017
imsDarkgreen	25600
imsDarkkhaki	7059389
imsDarkmagenta	9109643
imsDarkolivegreen	2845525
imsDarkorange	36095
imsDarkorchid	13382297
imsDarkred	139
imsDarksalmon	8034025
imsDarkseagreen	9419919
imsDarkslateblue	9125192
imsDarkslategray	5197615
imsDarkturquoise	13749760
imsDarkviolet	13828244
imsDeeppink	9633812
imsDeepskyblue	16760576
imsDimgray	6908265
imsDodgerblue	16748574
imsFirebrick	2237106
imsFloralwhite	15792895
imsForestgreen	2263842
imsFuchsia	16711935

imsGainsboro	14474460
imsGhostwhite	16775416
imsGold	55295
imsGoldenrod	2139610
imsGray	8421504
imsGreen	32768
imsGreenyellow	3145645
imsHoneydey	15794160
imsHotpink	11823615
imsIndianred	6053069
imsIndigo	8519755
imsIvory	15794176
imsKhaki	9234160
imsLavender	16744703
imsLavenderblush	16636917
imsLawngreen	64636
imsLemonchiffon	13499135
imsLightblue	15128749
imsLightcoral	8421616
imsLightcyan	16777184
imsLightgoldenrodyellow	13826810
imsLightgreen	9498256
imsLightgray	13882323
imsLightpink	12695295
imsLightsalmon	7446783
imsLightseagreen	11186720
imsLightskyblue	15388295
imsLightslategray	10061943
imsLightsteelblue	14599344
imsLightyellow	14745599
imsLime	65280
imsLimegreen	3329330
imsLinen	15134970
imsMagenta	16711935
imsMaroon	128
imsMediumaquamarine	11193702
imsMediumblue	13434880
imsMediumorchid	13850042
imsMediumpurple	14381203
imsMediumseagreen	7451452
imsMediumslateblue	15624315
imsMediumspringgreen	10156544
imsMediumturquoise	13422920

imsMediumvioletred	8721863
imsMidnightblue	7346457
imsMintcream	16449525
imsMistyrose	14804223
imsMoccasin	11920639
imsNavajowhite	11394815
imsNavy	8388608
imsOldlace	14087677
imsOlive	32896
imsOlivedrab	2330219
imsOrange	42495
imsOrangered	17919
imsOrchid	14053594
imsPalegoldenrod	11200750
imsPalegreen	10021784
imsPaleturquoise	15658671
imsPalevioletred	9662683
imsPapayawhip	14018047
imsPeachpuff	12180223
imsPeru	4163021
imsPink	13353215
imsPlim	14524637
imsPowderblue	15130800
imsPurple	8388736
imsRed	255
imsRosybrown	9408444
imsRoyalblue	14772545
imsSaddlebrown	1262987
imsSalmon	7504122
imsSandybrown	6333690
imsSeagreen	5737262
imsSeashell	15660543
imsSienna	2970272
imsSilver	12632256
imsSkyblue	15453831
imsSlateblue	13458026
imsSlategray	9470064
imsSnow	16448255
imsSpringgreen	8388352
imsSteelblue	11829830
imsTan	9221330
imsTeal	8421376
imsThistle	14204888
imsTomato	4678655
imsTurquoise	13688896
imsViolet	15631086
imsWheat	11788021
imsWhite	16777215
imsWhitesmoke	16119285
imsYellow	65535
imsYellowgreen	3329434

Returned Value

Long Color constants.

Example

```
imsGlobalColor.imsMediumaquamarine
```

imsGradientStyle constants

Description

Contains the gradient type.

Constants

imsBDiagonal	0	Backward diagonal gradient type.
imsFDiagonal	1	Forward diagonal gradient type.
imsHorizontal	2	Horizontal gradient type.
imsVertical	3	Vertical gradient type.

Returned Value

Long Gradient type.

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
mMap.Layers.Item(1).Renderer.Item(1).Symbol.GradientStyle = 1
mMap.Refresh
```

imsGroundStyle constants

Description

Contains the symbol type.

Constants

imsForeGround	0	Foreground type.
imsBackGround	1	Background type.

Returned Value

Long

Example

```
imsGroundStyle.imsForeGround
```

imsHAlignment constants

Description

Contains the horizontal alignment.

Constants

imsLeft	0	Left alignment.
imsCenter	1	Center alignment.
imsRight	2	Right alignment.

Returned Value

Long	Horizontal alignment.

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
Set al = Server.CreateObject("aims.AcetateLayer")
Al.visible = true
Set textobj = Server.CreateObject("aims.TextObject")
textobj.x = 25
textobj.y = 25
textobj.Label = "This is the Text!"
textobj.TextMarkerSymbol.Halignment = 1
al.Add textobj, 0
mMap.Layers.Add al
mMap.Refresh
```

imsHMLabels constants

Description

Determines how many labels are to be drawn to label a feature.

Constants

imsOne_label_per_name	0	Labels one label per feature name.
imsOne_label_per_shape	1	Labels one label per feature even if features with the same name exist.
imsOne_label_per_part	2	Labels all parts of a feature (in the case of multipart features).

Returned Value

Long

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
mMap.Layers.Item(1).Renderer.Item(2).HMLabels = 1
mMap.Refresh
```

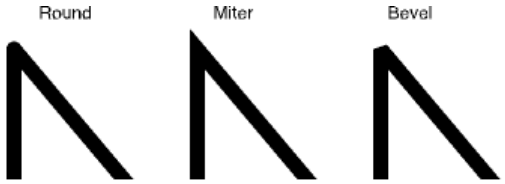
imsJoinStyle constants

Description

Contains the line intersect join types.

Constants

imsRound	Round join type.
imsMiter	Miter join type.
imsBevel	Bevel join type.



Returned Value

Long

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
mMap.Layers.Item(1).Renderer.Symbol.JoinStyle = 2
mMap.Refresh
```

imsLabelMode constants

Description

Contains the label mode for shield symbols.

Constants

imsFull	0	Takes the full label in the label field and puts it into the shield.
imsNumericOnly	1	Puts in the numbers from that label.

Returned Value

Long

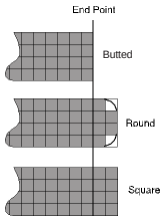
Example

```
Dim mArcIMSConnector
Dim mMap

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
```

```
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
mMap.Layers.Item(1).Renderer.Symbol.LabelMode = 1
mMap.Refresh
```



imsLineStyle constants

Description

Contains the line types.

Constants

imsSolid	0	Solid line type.
imsDash	1	Dash line type.
imsDot	2	Dot line type.
imsDash_dot	3	Dash–dot line type.
imsDash_dot_dot	4	Dash–dot–dot line type.

Returned Value

Long

Example

```
Dim mArcIMSConnector
Dim mMap

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400


mMap.Layers.Item(1).Renderer.Symbol.Style = 1
mMap.Refresh
```

imsLLPosition constants

Description

Determines where on the line to place the label.

Constants

imsPlaceNone	0	Do not place a label.
imsPlaceAbove	1	Place above the line.
imsPlaceBelow	2	Place below the line.
imsPlaceOnTop	3	Place on the line.
imsPlaceLeft	4	Place to the left of the line.
imsPlaceRight	5	Place to the right of the line.
imsPlaceAboveBelow	6	Place above or below the line.
imsPlaceLeftRight	7	Place at either end of the line.
imsPlaceInLine	8	Place anywhere on the line.
imsPlaceAtStart	9	Place at the beginning of the line.
imsPlaceAtEnd	10	Place at the end of the line.
imsPlaceAtEitherEnd	11	Place at the beginning or end of the line.
imsPlaceParallel	12	Place parallel to the line.
imsPlacePerpendicular	13	Place perpendicular to the line.
imsPlaceHorizontal	14	Place label so that it is always horizontal.
imsPlaceOnTopHorizontal	15	Place label on top of the line but always horizontal.

Returned Value

Long

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
mMap.Layers.Item(1).Renderer.Item(2).LLPosition = 12
mMap.Refresh
```

imsMapUnits constants

Description

Contains the map unit constants.

Constants

imsDecimalDegrees	0	Decimal degrees units.
imsMiles	1	Miles units.
imsFeet	2	Feet units.
imsKilometers	3	Kilometers units.
imsMeters	4	Meters units.
imsInches	5	Inches units.
imsCentimeters	6	Centimeters units.

Returned Value

Long

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
dim unit
unit = mMap.MapUnits
mMap.Refresh
```

imsMarkerType constants

Description

Contains point feature class layer marker types.

Constants

imsCircle	0	Circle marker type.
imsTriangle	1	Triangle marker type.
imsSquare	2	Square marker type.
imsCross	3	Cross marker type.
imsStar	4	Star marker type.

Returned Value

Long

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
mMap.Layers.Item(1).Renderer.Symbol.MarkerType = 3
mMap.Refresh
```

imsMethod constants

Description

Specifies the way a value in the data field is compared to the EXACT value. Use along with imsRange.imsExact.

Constants

imsIsContained 0
imsIsExact 1

Returned Value

Long

Example

```
Dim mValueMapRender
Dim mSimpleMarkerSymbol
Set mValueMapRenderer = Server.CreateObject ("aims.ValueMapRenderer")
mValueMapRenderer.LookupField = "COUNT"
Set mSimpleMarkerSymbol = Server.CreateObject ("aims.SimpleMarkerSymbol")
mResult = mValueMapRenderer.Add (mSimpleMarkerSymbol,
imsRange.imsExact,100,0,10,imsMethod.imsExact)
mMethod = mValueMapRenderer.Method(1)
```

imsPrintMode constants

Description

Contains label characters' printing mode types.

Constants

imsNone	0	No change is made.
imsTitleCaps	1	The first letter of each word in a label is uppercase, and everything else is lowercase.
imsAllUpper	2	All letters are uppercase.
imsAllLower	3	All letters are lowercase.

Returned Value

Long

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = true
Set textobj = Server.CreateObject("aims.TextObject")
textobj.x = 25
textobj.y = 25
textobj.Label = "This is the label!"
textobj.TextMarkerSymbol.Printmode = 3
al.add textobj, 0
mMap.Layers.Add al
mMap.Refresh
```

imsRange constants

Description

Defines the type of the symbol for ValueMapLabel and ValueMap renderers.

Constants

imsExact	0	Exact symbol constant.
imsRange	1	Range symbol constant.
imsOther	2	Other symbol constant.

Returned Value

Long

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
mMap.Layers.Item(1).Renderer.Item(1).Range(2) = 1
mMap.Refresh
```

imsScaleBarMode constants

Description

Specifies the method used to calculate the size and position of the scalebar.

Constants

imsCartesian	0
imsGeodesic	1

Returned Value

Long

Example

```
dim mScalebar
set mScalebar = CreateObject ("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = Arial
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0, degrees
mScalebar.ScaleUnits = 0, kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
myScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mScaleBar.BarTransparency = 1.0
mScaleBar.Mode = 0
mResult = Map.Layers.Add (AcetateLayer 1)
mResult = AcetateLayer1.Add (mScaleBar, 0)
```

imsShieldType constants

Description

Contains shield renderer symbol types.

Constants

imsInterstate	0	Interstate symbol type.
imsUSRoad	1	U.S. road symbol type.
imsRect	2	Rectangular symbol type.
imsOval	3	Ovular symbol type.
imsMexican	4	Mexican symbol type.

Returned Value

Long

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
mMap.Layers.Item(1).Renderer.Symbol.ShieldType = 2
mMap.Refresh
```

imsUnit constants

Description

Defines units for acetate objects.

Constants

imsPixel	0	Pixel units.
imsDatabase	1	Database units.

Returned Value

Long

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = true
Set pt = Server.CreateObject("aims.PointObject")
pt.x = -117.00
pt.y = 45.00
al.add pt, 1
mMap.Layers.Add al
mMap.Refresh
```

imsVAlignment constants

Description

Contains vertical alignment of labels or text.

Constants

imsTop	0	Top alignment.
imsCenter	1	Center alignment.
imsBottom	2	Bottom alignment.

Returned Value

Long

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 400
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = true
Set textobj = Server.CreateObject("aims.TextObject")
textobj.x = 25
textobj.y = 25
textobj.label = "This is the Label!"
textobj.TextMarkerSymbol.Valignment = 2
al.add textobj, 0
mMap.Layers.Add al
mMap.Refresh
```

imsWeight constants

Description

Prioritizes the importance of labels.

Constants

imsNo_weight	0	No weight constant.
imsMed_weight	1	Medium weight constant.
imsHigh_weight	2	High weight constant.

Returned Value

Long

Example

```
imsWeight.imsHigh_weight
```

Layers.Add method

Description
Adds a layer to the end of the layers collection.

Syntax
Layers.Add(layer as Object)

Argument
Layer as object Layer to add to the layers collection.

Returned Value
Boolean

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mLayer
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set mLayer = mMap.Layers.Item(1).Clone()
mMap.Layers.Add mLayer
mMap.Refresh
```

See Also
Buffer.asp
Create_Acetate_Layer.asp
Make_Spatial_Query.asp
Select_and_Highlight.asp

Layers.Clear method

Description
Clears the layers collection.

Syntax
Layers.Clear()

Arguments
None

Returned Value
None

Example
```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Layers.Clear()
mMap.Refresh
```

Layers.Count property

Description
Number of layers in the layers collection. Read only.

Type
Long

Example
```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mCount = mMap.Layers.Count
Response.write CStr(mCount)
```

See Also
Show_Map.asp

Layers.Create method

Description

Creates a FeatureLayer or an ImageLayer based on the information from parameter "Workspace". For a FeatureLayer, "Workspace" should be SDEWorkspace or ShapeWorkspace; for an image layer, "Workspace" should be ImageWorkspace. For a newly created FeatureLayer, the recordset property will be Nothing.

Syntax

Layers.Create(Workspace as Object)

Argument

Workspace	Workspace object.

Returned Value

Object	Just created layer. It can be a FeatureLayer or ImageLayer according to the type of workspace passed as the parameter.

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mLayer
Dim mShapeWorkspace

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mShapeWorkspace = Server.CreateObject("aims.ShapeWorkspace")
mShapeWorkspace.Name = "Country"
mShapeWorkspace.Directory = "D:\EsriData\World"
mShapeWorkspace.FeatureClass = imsFeatureClass.imsPolygon

Set mLayer = mMap.Layers.Create(mShapeWorkspace)
mMap.Layers.Add mLayer
mMap.Refresh
```

Layers.CreateBuffer method

Description

Creates a new feature class layer that contains a constructed buffer zone. TargetLayer will contain only objects that are located in the buffer zone.

Syntax

Layers.CreateBuffer(LayerFilter as Filter, Targetlayer as FeatureLayer)

Arguments

Filter	A base filter. Buffer zone will be constructed near objects that are located in the filter.
TargetLayer	An optional parameter. TargetLayer will contain only objects that are located in the buffer zone.

Returned Value

FeatureLayer	New feature class layer that contains constructed buffer zone.

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mLayer
Dim mPointObject
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set mPointObject = Server.CreateObject("aims.PointObject")
mPointObject.X = -122.7
mPointObject.Y = -37.21
mMap.Layers.Item(1).Filter.AddGObject(mPointObject)
mMap.Layers.Item(1).Filter.Buffer = 0.5
Set mLayer = mMap.Layers.CreateBuffer(mMap.Layers.Item(1).Filter)
mMap.Layers.Add  mLayer
mMap.Refresh
```

See Also

Buffer.asp

Layers.CreateHighlight method

Description

Creates a new feature class layer that contains specified objects only. The recordset of the new layer will contain only highlighted objects. By default, a renderer is specified (yellow, 50 percent transparent).

Syntax

FeatureLayer Layers.CreateHighlight(ActiveLayer, FieldName, Values)

Argument

ActiveLayer	FeatureLayer	Active layer.
FieldName	String	Name of the field.
Values	String	Values to highlight delimited by commas.

Returned Value

FeatureLayer	New feature class layer.

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mLayer

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mLayer = mMap.Layers.CreateHighlight(mMap.Layers.Item(1), "CNTRY_NAME",
"'USA', 'RUSSIA'")

mMap.Layers.Add mLayer
```

Layers.Find method

Description

Lets you pass a layer identifier and have a layer index returned.

Syntax

Layers.Find(LayerId as String) as Integer

Argument

String Provides the layer identifier for which you want an index.

Returned Value

Integer Returns the index for the layer.

Example

This example shows how to use this method to return a layer you want, where “city” is the identifier of the layer you want.

```
Dim layer, index

index=mMap.Layers.Find("city")
Set layer=mMap.Layers.item(index)
```

Layers.Insert method

Description

Inserts the layer at the specified position.

Syntax

Layers.Insert(layer as object, index as long)

Arguments

Layer	Object	Layer.
Index	Long	Position to insert.

Returned Value

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mLayer

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mLayer = mMap.Layers.Item(1).Clone()
mMap.Layers.Insert  mLayer, 3
mMap.Refresh
```

Layers.Item method

Description
Returns the layer by index.

Syntax
Layers.Item(index as Long)

Argument
Index	Long	Description.

Returned Value
Object	Returns the layer referenced by the index.

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mLayer

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mLayer = mMap.Layers.Item(1)
Response.write mLayer.Name
```

See Also
Show_Map.asp

Layers.MoveTo method

Description
Moves the layer.

Syntax
Layers.MoveTo(fromIndex as Long, toIndex as Long)

Arguments

fromIndex	Long	Start position.
toIndex	Long	End position.

Returned Value
Boolean

Example
```
Dim mArcIMSConnector
Dim mMap

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

mMap.Layers.MoveTo 2, 4
mMap.Refresh
```

See Also
Change_layers_order.asp

Layers.MoveToBottom method

Description

Moves the layer to the beginning of the layers collection and places the layer under other layers.

Syntax

Layers.MoveToBottom(index as Long)

Argument

Index	Long	Index to reference to the layer.

Returned Value

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

mMap.Layers.MoveToBottom 2
mMap.Refresh
```

See Also

Layers.MoveToTop

Layers.MoveToTop method

Description

Moves the layer to the end of the layers collection and displays the layer on top of layers.

Syntax

Layers.MoveToTop(index as Long)

Argument

Index	Long	Index to reference to the layer.

Returned Value

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.Init mArcIMSConnector, "IMSMapService"

mMap.Layers.MoveToTop 2

mMap.Refresh
```

See Also

Layers.MoveToBottom

Layers.NoDefault property

Description

Setting the value to True limits the layers that are extracted to the visible feature layers. When NoDefault is False, all the visible feature layers will be extracted. For both True and False, layers not visible at that scale defined by their ScaleDependentRenderer will not be extracted. Write/Read.

Type

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap
Dim boolExtract
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.BackColor = 128
mMap.TransColor = 128
boolExtract=mMap.isMapExtractable
if boolExtract then
mMap.Extract = True
end If
mMap.Layers.Item(2).Visible = False
mMap.Layers.NoDefault = True
mMap.Refresh
```

Layers.Remove method

Description

Removes the layer from the layers collection by index.

Syntax

Layers.Remove(index as Long)

Argument

Index	Long	Index to reference to layer.

Returned Value

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

mMap.Layers.Remove 1
mMap.Refresh
```

See Also

Buffer.asp
Create_Acetate_Layer.asp
Select_and_Highlight.asp

Legend.Autoextent property

Description
If True, automatically extends the legend vertically past the size specified in height, if needed. Write/Read.

Type
Boolean

Example
```
Dim mLegend
Set mLegend = Server.CreateObject("aims.Legend")
mLegend.Autoextent = True
mURL = mLegend.URL
Response.write mURL
```

Legend.Background property

Description
Contains the legend's background color. Write/Read.

Type
Long

Example
```
Dim mLegend
Dim mURL
Set mLegend = Server.CreateObject("aims.Legend")
mLegend.Background = 255
mURL = mLegend.URL
Response.write mURL
```

See Also
Change_layer_visibility.asp

Legend.CanSplit property

Description
Permits the splitting of valuemap layers. Write/Read.

Type
Boolean

Example
```
Dim mLegend
Dim mURL
Set mLegend = Server.CreateObject("aims.Legend")
mLegend.CanSplit = True
mURL = mLegend.URL
Response.write mURL
```

Legend.Cellspacing property

Description
Defines the number of pixels to pad between entries. Write/Read.

Type
Long

Example
```
Dim mLegend
Dim mURL
Set mLegend = Server.CreateObject("aims.Legend")
mLegend.Cellspacing = 10
mURL = mLegend.URL
Response.write mURL
```

See Also
Change_layer_visibility.asp

Legend.Columns property

Description
Defines the number of columns the splits will apply to in the valuemap. Write/Read.

Type
Long

Example
```
Dim mLegend
Dim mURL
Set mLegend = Server.CreateObject("aims.Legend")
mLegend.Columns = 10
mURL = mLegend.URL
Response.write mURL
```

Legend.Display property

Description
Turns the legend on or off. Write/Read.

Type
Boolean

Example
```
Dim mLegend
Dim mURL
Set mLegend = Server.CreateObject("aims.Legend")
mLegend.Display = True
mURL = mLegend.URL
Response.write mURL
```

Legend.Font property

Description
Title's font name. Write/Read.

Type
String

Example
```
Dim mLegend
Dim mURL
Set mLegend = Server.CreateObject("aims.Legend")
mLegend.Font = "Arial"
mURL = mLegend.URL
Response.write mURL
```

Legend.Height property

Description
Contains the legend height in pixels. Write/Read.

Type
Long

Example
```
Dim mLegend
Dim mURL
Set mLegend = Server.CreateObject("aims.Legend")
mLegend.Height = 10
mURL = mLegend.URL
Response.write mURL
```

Legend.LayerFontSize property

Description
Contains the layer name's font size. Write/Read.

Type
Long

Example
```
Dim mLegend
Dim mURL
Set mLegend = Server.CreateObject("aims.Legend")
mLegend.LayerFontSize = 10
mURL = mLegend.URL
Response.write mURL
```

Legend.ReverseOrder property

Description
Reverses the order of the layers. Write/Read.

Type
Boolean

Example
```
Dim mLegend
Dim mURL
Set mLegend = Server.CreateObject("aims.Legend")
mLegend.ReverseOrder = True
mURL = mLegend.URL
Response.Write mURL
```

Legend.SplitText property

Description
Displays in the bottom of every column that has been split for the valuemap. Write/Read.

Type
String

Example
```
Dim mLegend
Dim mURL
Set mLegend = Server.CreateObject("aims.Legend")
mLegend.SplitText = "Split text"
mURL = mLegend.URL
Response.write mURL
```

Legend.SwatchHeight property

Description
Height of the symbol swatch in pixels.

Type
Long

Example
```
Dim mLegend
Dim mURL
Set mLegend = Server.CreateObject("aims.Legend")
mLegend.SwatchHeight = 10
mURL = mLegend.URL
Response.write mURL
```

Legend.SwatchWidth property

Description
Width of the symbol swatch in pixels.

Type
Long

Example
```
Dim mLegend
Dim mURL
Set mLegend = Server.CreateObject("aims.Legend")
mLegend.SwatchWidth = 10
mURL = mLegend.URL
Response.write mURL
```

Legend.Title property

Description
Contains the title of the entire legend. Write/Read.

Type
String

Example
```
Dim mLegend
Dim mURL
Set mLegend = Server.CreateObject("aims.Legend")
mLegend.Title = "Legend"
mURL = mLegend.URL
Response.write mURL
```

See Also
Change_layer_visibility.asp

Legend.TitleFontSize property

Description
Contains the font size of the title to be placed on the legend. Write/Read.

Type
Long

Example
```
Dim mLegend
Dim mURL
Set mLegend = Server.CreateObject("aims.Legend")
mLegend.TitleFontSize = 10
mURL = mLegend.URL
Response.write mURL
```

See Also
Change_layer_visibility.asp

Legend.Transcolor property

Description
Contains transparency information for the legend. Write/Read.

Type
Long

Example
```
Dim mLegend
Dim mURL
Set mLegend = Server.CreateObject("aims.Legend")
mLegend.Transcolor = 255
mURL = mLegend.URL
Response.write mURL
```

Legend.Url property

Description
Contains the HTTP address to the legend graphic. Write/Read.

Type
String

Example
```
Dim mLegend
Dim mURL
Set mLegend = Server.CreateObject("aims.Legend")
mURL = mLegend.URL
Response.write mURL
```

See Also
Change_layer_visibility.asp

Legend.ValueFontSize property

Description
Contains the font size for unique values of layers. Write/Read.

Type
Long

Example
```
Dim mLegend
Dim mURL
Set mLegend = Server.CreateObject("aims.Legend")
mLegend.ValueFontSize = 10
mURL = mLegend.URL
Response.write mURL
```

Legend.Width property

Description

Legend width in pixels. Write/Read.

Type

Long

Example

```
Dim mLegend
Dim mURL

Set mLegend = Server.CreateObject("aims.Legend")
mLegend.Width = 10

mURL = mLegend.URL
Response.write mURL
```

LineObject.Id property

Description
String identifier for the line. Write/Read.

Type
Long

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mLineObject
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set mLineObject = Server.CreateObject("aims.LineObject")
mId = mLineObject.Id
Response.write mId
```

LineObject.Name property

Description
Line's name. Write/Read.

Type
String

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mLineObject
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set mLineObject = Server.CreateObject("aims.LineObject")
mLineObject.Name = "The Line"
Response.write mLineObject.Name
```

LineObject.Parts property

Description

Contains parts of the line. Each part is a simple line. Read only.

Type

Parts

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mLineObject
Dim mParts
Dim mPointObject
Dim mPoints
Dim al
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = True
Set mPoints = Server.CreateObject("aims.Points")
Set mPointObject1 = Server.CreateObject("aims.PointObject")
Set mPointObject2 = Server.CreateObject("aims.PointObject")
mPointObject1.x = -125.0
mPointObject1.y = 45.0
mPointObject2.x = -120.00
mPointObject2.y = 50.00
mPoints.Add mPointObject1
mPoints.Add mPointObject2
Set mLineObject = Server.CreateObject("aims.LineObject")
Set mParts = mLineObject.Parts
mParts.Add mPoints
al.Add mLineObject, 1
```

LineObject.Symbol property

Description

Specifies how to depict the line using any line symbol. Write/Read.

Type

Object

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mLineObject
Dim mParts
Dim mPointObject
Dim mPoints
Dim al
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = True
Set mPoints = Server.CreateObject("aims.Points")
Set mPointObject1 = Server.CreateObject("aims.PointObject")
Set mPointObject2 = Server.CreateObject("aims.PointObject")
mPointObject1.x = -125.0
mPointObject1.y = 45.0
mPointObject2.x = -120.00
mPointObject2.y = 50.00
mPoints.Add mPointObject
mPoints.Add mPointObject2

Set mLineObject = Server.CreateObject("aims.LineObject")
mLineObject.Parts.Add mPoints
mLineObject.Symbol.Color = 255
al.Add mLineObject, 1
mMap.Layers.Add al
mMap.Refresh
```

See Also

HashLineSymbol properties and methods
SimpleLineSymbol properties and methods

Map.BackColor property

Description
Background color of the map. Use imsColor and imsGlobalColor constants. Write/Read.

Type
Long

Example
```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.BackColor = 128
mMap.Refresh
```

See Also
imsColor
imsGlobalColor
Show_Map.asp

Map.CenterAt method

Description
Sets map center to the specified point. Coordination unit is pixels.

Syntax
Map.CenterAt(X as Long, Y as Long)

Arguments

X	Long	X coordinate of the center in pixels.
Y	Long	Y coordinate of the center in pixels.

Returned Value
None

Examples
```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.CenterAt 100, 100
mMap.Refresh
```

See Also
Zoom_In.asp

Map.DoPan method

Description

Shifts the current extent in the specified direction.

Syntax

Map.DoPan (direction as Long, step as Double)

Arguments

Direction	Long	Specifies direction. Use imsDirection constant.
Step	Double	Specifies the step for pan.

Returned Value

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.DoPan imsDirection.imsNorth, 0.5
mMap.Refresh
```

See Also

imsDirection

Map.DoZoom method

Description
Controls zooming of the map.

Syntax
Map.DoZoom(ScaleFactor as Long)

Argument
ScaleFactor as Long Scale factor > 0 for zoom in. Scale factor < 0 for zoom out.

Returned Value
None

Example
```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.DoZoom 1
mMap.Refresh
```

See Also
Map.DoPan
Map.DoZoomToExtent
Map.DoZoomToFeatures
Map.DoZoomToFullExtent
Zoom_In.asp

Map.DoZoomToExtent method

Description
Sets the map extent to the specified extent.

Syntax
Map.DoZoomToExtent(Extent as Envelope)

Argument
Extent as envelope The required extent.

Returned Value
Boolean

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mExtent
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
Set mExtent = Server.CreateObject("aims.Envelope")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mExtent.XMin = 112
mExtent.YMin = -8
mExtent.XMax = 203
mExtent.YMax = 64
mMap.DoZoomToExtent mExtent
mMap.Refresh
```

See Also
Map.DoPan
Map.DoZoom
Map.DoZoomToFeatures
Map.DoZoomToFullExtent

Map.DoZoomToFeatures method

Description

This method has multiple functions. It first queries the ActiveLayer, if the value of a field, specified in FieldName, is one of the Values. Then, the map is zoomed to the minimum envelope that can contain all the features satisfying the query. In the meantime, the recordset of the ActiveLayer is updated to display only features satisfying the query. Note that if the qualified feature is only one point, the map is not zoomed since the envelope of one point is 0.

Syntax

Map.DoZoomToFeatures(ActiveLayer, FieldName, Values)

Arguments

ActiveLayer as FeatureLayer	Active layer.
FieldName as string	Specifies the field name.
Values as string	Values for search delimited by comma.

Returned Value

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.DoZoomToFeatures mMap.Layers.Item(2), "CNTRY_NAME", "'USA','Russia'"
mMap.Refresh
```

See Also

Map.DoPan
Map.DoZoom
Map.DoZoomToExtent
Map.DoZoomToFullExtent

Map.DoZoomToFullExtent method

Description
Sets the map extent to the initial extent defined in the configuration file.

Syntax
Map.DoZoomToFullExtent()

Arguments
None

Returned Value
Boolean

Example
```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.DoZoomToFullExtent()
mMap.Refresh
```

See Also
Map.DoPan
Map.DoZoom
Map.DoZoomToExtent
Map.DoZoomToFeatures

Map.Extent property

Description
Current extent of the map. Read only.

Type
Object

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mExtent
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set mExtent = mMap.Extent
```

See Also
Buffer.asp
Change_layer_visibility.asp
Change_layers_order.asp
Envelope

Map.Extract property

Description
When Map.isMapExtractable returns true, set Map.Extract to true to enable layer extraction; if the Map.Extract is set to false, no layer will be extracted. Write only.

Type
Boolean

Example
```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
if mMap.isMapExtractable then
   mMap.Extract = True
   mMap.Refresh
   Response.Write "File for extracted layers: " & mMap.GetExtractUrl()
else
   Response.Write "Not Extractable"
end if
```

Map.FromMapPoint method

Description

Converts coordinates from database units to pixel units.

Syntax

PointObject Map.FromMapPoint(X as Double,Y as Double)

Arguments

X	Double	X coordinate in database units.
Y	Double	Y coordinate in database units.

Returned Value

PointObject	PointObject contains converted coordinates in pixel units.

Example

```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 300
mMap.Refresh
Set mPointObject=mMap.FromMapPoint(mMap.Extent.XMax, mMap.Extent.YMin)
Response.Write "In pixels " & mPointObject.x & ", " & mPointObject.y
```

See Also

Map.ToMapPoint

Map.GetExtractFile method

Description

Returns a string value containing the physical path to the zipped file having the extracted layers.

Syntax

Map.GetExtractFile()

Arguments

None

Returned Value

String value containing the physical path to the zipped file having the extracted layers.

Example

```
Dim mArcIMSConnector
Dim mMap
Dim boolExtract
Dim strExtractUrl
Dim strExtractFilePath
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.BackColor = 128
mMap.TransColor = 128
boolExtract = mMap.isMapExtractable
if boolExtract then
mMap.Extract = True
end If
mMap.Refresh
strExtractFilePath = mMap.GetExtractFile()
```

Map.GetExtractUrl method

Description
Returns a string value containing the URL to the zipped file having the extracted layers.

Syntax
Map.GetExtractUrl()

Arguments
None

Returned Value
String value containing the URL to the zipped file having the extracted layers.

Example
```
Dim mArcIMSConnector
Dim mMap
Dim boolExtract
Dim strExtractUrl Dim strExtractFilePath
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.BackColor = 128
mMap.TransColor = 128
boolExtract = mMap.isMapExtractable
if boolExtract then
mMap.Extract = True
end If
mMap.Refresh
strExtractUrl = 1mMap.GetExtractUrl()
```

Map.GetImageAsUrl method

Description
Returns the URL of the generated image.

Syntax
Map.GetImageAsURL()

Arguments
None

Returned Value
String Address of the picture.

Example
```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Refresh
Response.write "The map image is at: " & mMap.GetImageAsUrl()
```

See Also
Show_Map.asp

Map.Height property

Description
Height of the map in pixels. Write/Read.

Type
Long

Example
```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Height = 300
```

See Also
Map.Width
Show_Map.asp

Map.InitMap method

Description

Initializes the service and loads the necessary renderer, recordset, and geocoding data.

Syntax

Map.InitMap(Connector, ServiceName, LoadRenderer, LoadRecordset, LoadGeocode)

Arguments

Connector	ArcIMSConnector	Connector object that contains all connection information.
ServiceName	String	Name of the ArcIMS service.
LoadRenderer	Boolean	Optional parameter. If False, the layers will not work with the Renderer, and the Renderer property of all feature class layers will be Nothing. Default is True.
LoadRecordset	Boolean	Optional parameter. If False, the layers will not work with recordset, and the Recordset property of all feature class layers will be Nothing. Default is True.
LoadGeocode	Boolean	Optional parameter. If False, the connector won't send an additional request to download geocoding data, and the AddressMatchInputs property of all feature class layers will be Nothing. Default is True.

Returned Value

Boolean	To get the last error, use the ArcIMSConnector.GetLastError() method.

Example

```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService", false, true, false
```

See Also

ArcIMSConnector
ArcIMSConnector.GetLastError
Show_Map.asp

Map.isMapExtractable property

Description

ArcIMS sets Map.isMapExtractable to True if the configuration file has at least one layer with the Type attribute of the <EXTENSION> tag set to Extract. If it's set to True, you can set Map.Extract to True, which permits your users to extract any layer(s) in the ArcIMS service, whether the layer has the Type attribute of <EXTENSION> set to Extract or not. Read only.

Type

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap
Dim boolExtract
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.BackColor = 128
mMap.TransColor=128
boolExtract=mMap.isMapExtractable
mMap.Refresh
```

Map.Legend property

Description

Contains all legend parameters. Write/Read.

Type

Legend

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mLegend

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mLegend = mMap.Legend
Response.write mLegend.URL
```

See Also

Change_layer_visibility.asp
Legend properties and methods

Map.MapDraw property

Description
If set to False, disables the generation of a map image on a map refresh. Indicates whether a map image is generated. Write/Read.

Type
Boolean

Example
```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.BackColor = 128
mMap.MapDraw = False
mMap.Refresh
```

Note
Refer to the <DRAW> tag in the *ArcXML Programmer's Reference Guide*.

Map.Refresh method

Description
Refreshes the map.

Syntax
Map.Refresh()

Arguments
None

Returned Value
Boolean To get last error description, use ArcIMSConnector.GetLastError().

Example
```
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.Width = 500
mMap.Height = 300
mMap.Refresh
```

See Also
ArcIMSConnector.GetLastError
Show_Map.asp

Map.ToMapPoint method

Description

Converts coordinates from pixel units to database units.

Syntax

Map.ToMapPoint(X as Long,Y as Long)

Arguments

X	Long	X coordinate in pixels.
Y	Long	Y coordinate in pixels.

Returned Value

PointObject	PointObject contains converted coordinates in database units.

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mPointObject

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mPointObject = mMap.ToMapPoint(100, 100)
Response.write mPointObject.x & "," & mPointObject.y
```

See Also

Identify.asp
Map.FromMapPoint
Select_and_Highlight.asp

Map.TransColor property

Description

Transparency color of the map. In order to set the background of an Image Service as transparent, BackColor and TransColor should be set to the same value. Only supported for .gif and .png output formats. Use imsColor and imsGlobalColor constants. Write/Read.

Type

Long

Example

```
Dim mArcIMSConnector
Dim mMap
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
mMap.BackColor = 128
mMap.TransColor = 128
mMap.Refresh
```

See Also

imsColor
imsGlobalColor

Note

Refer to the <BACKGROUND> tag in the *ArcXML Programmer's Reference Guide*.

Map.Width property

Description
Width of the map in pixels. Write/Read.

Type
Long

Example
```
Dim mArcIMSConnector
Dim mMap


Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

mMap.Width = 500
```

See Also
Map.Height
Show_Map.asp

NorthArrowObject.Angle property

Description

Angle in simple degrees at which to turn the North arrow. Write/Read.

Type

Double

Example

```
Dim mArcIMSConnector
Dim mMap
Dim al
Dim mNorthArrowObject
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = true

Set mNorthArrowObject = Server.CreateObject("aims.NorthArrowObject")

mNorthArrowObject.Angle = 31
mNorthArrowObject.x = 25
mNorthArrowObject.y = 25
mNorthArrowObject.Size = 15
mNorthArrowObject.ArrowType = 2
al.Add mNorthArrowObject, 0
mMap.Layers.Add al
mMap.Refresh
```

NorthArrowObject.Antialiasing property

Description

Turns antialiasing on/off. Antialiasing allows lines to appear smooth by coloring adjacent pixels in such a way as to give the edge of a line a more gradual fade into the background. Write/Read.

Type

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap
Dim al
Dim mNorthArrowObject
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = true
Set mNorthArrowObject = Server.CreateObject("aims.NorthArrowObject")
mNorthArrowObject.Angle = 31
mNorthArrowObject.Antialiasing = true
mNorthArrowObject.x = 25
mNorthArrowObject.y = 25
mNorthArrowObject.Size = 15
mNorthArrowObject.ArrowType = 2
al.Add mNorthArrowObject, 0
mMap.Layers.Add al
mMap.Refresh
```

NorthArrowObject.ArrowType property

Description

Arrow type. Use imsArrowType constants. Write/Read.

Type

Long

Example

```
Dim mArcIMSConnector
Dim mMap
Dim al
Dim mNorthArrowObject
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = true
Set mNorthArrowObject = Server.CreateObject("aims.NorthArrowObject")

mNorthArrowObject.Angle = 31
mNorthArrowObject.x = 25
mNorthArrowObject.y = 25
mNorthArrowObject.Size = 15
mNorthArrowObject.ArrowType = 2
al.Add mNorthArrowObject, 0
mMap.Layers.Add al
mMap.Refresh
```

See Also

Create_Acetate_Layer.asp
imsArrowType

NorthArrowObject.Id property

Description

String identifier for the NorthArrowObject. Write/Read.

Type

String

Example

```
Dim mArcIMSConnector
Dim mMap
Dim al
Dim mNorthArrowObject
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = true
Set mNorthArrowObject = Server.CreateObject("aims.NorthArrowObject")
mNorthArrowObject.Angle = 31
mNorthArrowObject.x = 25
mNorthArrowObject.y = 25
mNorthArrowObject.Size = 15
mNorthArrowObject.ArrowType = 2
mNorthArrowObject.Id = "NARROW:1976"
al.Add mNorthArrowObject, 0
mMap.Layers.Add al
mMap.Refresh
```

NorthArrowObject.Name property

Description

The name you give the north arrow object. Write/Read.

Type

String

Example

```
Dim mArcIMSConnector
Dim mMap
Dim al
Dim mNorthArrowObject
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = true
Set mNorthArrowObject = Server.CreateObject("aims.NorthArrowObject")
mNorthArrowObject.Angle = 31
mNorthArrowObject.x = 25
mNorthArrowObject.y = 25
mNorthArrowObject.Size = 15
mNorthArrowObject.ArrowType = 2
mNorthArrowObject.Name = "NorthArrow 5"
al.Add mNorthArrowObject, 0
mMap.Layers.Add al
mMap.Refresh
```

NorthArrowObject.Outline property

Description

Outline color. Write/Read.

Type

Long

Example

```
Dim mArcIMSConnector
Dim mMap
Dim al
Dim mNorthArrowObject
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = true
Set mNorthArrowObject = Server.CreateObject("aims.NorthArrowObject")
mNorthArrowObject.Angle = 31
mNorthArrowObject.x = 25
mNorthArrowObject.y = 25
mNorthArrowObject.Size = 15
mNorthArrowObject.ArrowType = 2
mNorthArrowObject.Outline = 255
al.Add mNorthArrowObject, 0
mMap.Layers.Add al
mMap.Refresh
```

See Also

imsColor

NorthArrowObject.Overlap property

Description

Determines if labels can overlap this symbol. When True, labels can overlap. When False, labels cannot overlap the symbol. Write/Read.

Type

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap
Dim al
Dim mNorthArrowObject
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = true
Set mNorthArrowObject = Server.CreateObject("aims.NorthArrowObject")
mNorthArrowObject.Angle = 31
mNorthArrowObject.x = 25
mNorthArrowObject.y = 25
mNorthArrowObject.Size = 15
mNorthArrowObject.ArrowType = 2
mNorthArrowObject.Overlap = False
al.Add mNorthArrowObject, 0
mMap.Layers.Add al
mMap.Refresh
```

NorthArrowObject.Shadow property

Description

Shadow color. Write/Read.

Type

Long

Example

```
Dim mArcIMSConnector
Dim mMap
Dim al
Dim mNorthArrowObject
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = true
Set mNorthArrowObject = Server.CreateObject("aims.NorthArrowObject")
mNorthArrowObject.Angle = 31
mNorthArrowObject.x = 25
mNorthArrowObject.y = 25
mNorthArrowObject.Size = 15
mNorthArrowObject.ArrowType = 2
mNorthArrowObject.Shadow = 0
al.Add mNorthArrowObject, 0
mMap.Layers.Add al
mMap.Refresh
```

See Also

imsColor

NorthArrowObject.Size property

Description
Arrow size. Write/Read.

Type
Long

Example
```
Dim mArcIMSConnector
Dim mMap
Dim al
Dim mNorthArrowObject
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = true
Set mNorthArrowObject = Server.CreateObject("aims.NorthArrowObject")
mNorthArrowObject.Angle = 31
mNorthArrowObject.x = 25
mNorthArrowObject.y = 25
mNorthArrowObject.Size = 15
mNorthArrowObject.ArrowType = 2
al.Add mNorthArrowObject, 0
mMap.Layers.Add al
mMap.Refresh
```

See Also
Create_Acetate_Layer.asp

NorthArrowObject.Transparency property

Description

Transparency for the North arrow. Write/Read.

Type

Double

Example

```
Dim mArcIMSConnector
Dim mMap
Dim al
Dim mNorthArrowObject
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = true

Set mNorthArrowObject = Server.CreateObject("aims.NorthArrowObject")

mNorthArrowObject.Angle = 31
mNorthArrowObject.x = 25
mNorthArrowObject.y = 25
mNorthArrowObject.Size = 15
mNorthArrowObject.ArrowType = 2
mNorthArrowObject.Transparency = 0.7
al.Add mNorthArrowObject, 0
mMap.Layers.Add al
mMap.Refresh
```

See Also

Create_Acetate_Layer.asp

NorthArrowObject.X property

Description

X coordinate of arrow location. Write/Read.

Type

Double

Example

```
Dim mArcIMSConnector
Dim mMap
Dim al
Dim mNorthArrowObject
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = true
Set mNorthArrowObject = Server.CreateObject("aims.NorthArrowObject")
mNorthArrowObject.Angle = 31
mNorthArrowObject.x = 25
mNorthArrowObject.y = 25
mNorthArrowObject.Size = 15
mNorthArrowObject.ArrowType = 2
al.Add mNorthArrowObject, 0
mMap.Layers.Add al
mMap.Refresh
```

See Also

Create_Acetate_Layer.asp

NorthArrowObject.Y property

Description

Y coordinate of the arrow location. Write/Read.

Type

Double

Example

```
Dim mArcIMSConnector
Dim mMap
Dim al
Dim mNorthArrowObject
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = true
Set mNorthArrowObject = Server.CreateObject("aims.NorthArrowObject")
mNorthArrowObject.Angle = 31
mNorthArrowObject.x = 25
mNorthArrowObject.y = 25
mNorthArrowObject.Size = 15
mNorthArrowObject.ArrowType = 2
al.Add mNorthArrowObject, 0
mMap.Layers.Add al
mMap.Refresh
```

See Also

Create_Acetate_Layer.asp

Parts.Add method

Description
Adds a Points object to the Parts collection.

Syntax
Parts.Add(Points)

Argument
Points	Points	Points object to add.

Returned Value
None

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mParts
Dim mPoints
Dim mPoint
Dim al
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set mParts = Server.CreateObject("aims.Parts")
Set mPoints = Server.CreateObject("aims.Points")
Set mPoint = Server.CreateObject("aims.PointObject")
mPoint.x = -125.00
mPoint.y = 45.00
mPoints.Add mPoint
mParts.Add(mPoints)
Set al = Server.CreateObject("aims.AcetateLayer")
al.visible = true
al.Add mPoints, 1
mMap.Layers.Add al
mMap.Refresh
```

Parts.Clear method

Description
Removes all Points objects from the Parts collection.

Syntax
Parts.Clear()

Arguments
None

Returned Value
None

Example
```
Dim mParts
Dim mPoints
Set mParts = Server.CreateObject("aims.Parts")
Set mPoints = Server.CreateObject("aims.Points")
mParts.Add(mPoints)
Response.Write mParts.Count()
mParts.Clear()
Response.Write mParts.Count()
```

Parts.Count property

Description
Number of Points objects in the Parts collection. Read only.

Type
Long

Example
```
Dim mParts
Dim mPoints
Set mParts = Server.CreateObject("aims.Parts")
Set mPoints = Server.CreateObject("aims.Points")
mParts.Add(mPoints)
mCount = mParts.Count
```

Parts.Insert method

Description
Inserts a Points object at the specified position.

Syntax
Parts.Insert(index as Long, Points as Points)

Arguments

Index	Long	Points object position.
Points	Points	Points object to insert.

Returned Value
None

Example
```
Dim mParts
Dim mPoints
Set mParts = Server.CreateObject("aims.Parts")
Set mPoints = Server.CreateObject("aims.Points")
mParts.Insert(1, mPoints)
```

Parts.Item method

Description
Returns the Points object by index.

Syntax
Parts.Item(index as Long)

Argument

Index	Long	Points object position.

Returned Value

Points	Description.

Example
```
Dim mParts
Dim mPoints
Set mParts = Server.CreateObject("aims.Parts")
Set mPoints = Server.CreateObject("aims.Points")
mParts.Add(mPoints)
Set mPoints = mParts.Item(1)
```

Parts.Remove method

Description
Removes the Points object from the collection.

Syntax
Parts.Remove(index as Long)

Argument

Index	Long	Points object position.

Returned Value
None

Example
```
Dim mParts
Dim mPoints
Set mParts = Server.CreateObject("aims.Parts")
Set mPoints = Server.CreateObject("aims.Points")
mParts.Add(mPoints)
mParts.Remove(1)
```

Parts.Replace method

Description
Replaces the Points object.

Syntax
Parts.Replace(index asLong, Points as Points)

Arguments

Index	Long	Object’s position.
Points	Points	New Points object.

Returned Value
None

Example
```
Dim mParts
Dim mPoints
Set mParts = Server.CreateObject("aims.Parts")
Set mPoints = Server.CreateObject("aims.Points")
mParts.Add(mPoints)
mParts.Replace(1, mPoints)
```

PointObject.Id property

Description
String identifier for a PointObject. The default value is Point. Write/Read.

Type
String

Example
```
Dim mPointObject
Set mPointObject = Server.CreateObject("aims.PointObject")
mId = mPointObject.Id
```

PointObject.Name property

Description
String name for a PointObject. The default value is Point. Write/Read.

Type
String

Example
```
Dim mPointObject
Set mPointObject = Server.CreateObject("aims.PointObject")
mName = mPointObject.Name
```

PointObject.Symbol property

Description
Symbol object of the PointObject. The symbol object type is either SimpleMarkerSymbol, TrueTypeMarkerSymbol, or RasterMarkerSymbol. The default object is SimpleMarkerSymbol. Write/Read.

Type
Object

Example
```
Dim mPointObject
Dim mSymbol
Set mPointObject = Server.CreateObject("aims.PointObject")
Set mSymbol = mPointObject.Symbol
```

See Also
RasterMarkerSymbol methods and properties
SimpleMarkerSymbol methods and properties
TrueTypeMarkerSymbol methods and properties

PointObject.X property

Description

X coordinate of the point. The coordinate units can be in pixel or map units. The default value is 0. Write/Read.

Type

Double

Example

```
Dim mPointObject
Set mPointObject = Server.CreateObject("aims.PointObject")
mPointObject.Y = -37.32
mPointObject.X = -122.7
```

See Also

Identify.asp
PointObject.Y
Select_and Highlight.asp

PointObject.Y property

Description

Y coordinate of the point. The coordinate units can be in pixel or map units. The default value is 0. Write/Read.

Type

Double

Example

```
Dim mPointObject
Set mPointObject = Server.CreateObject("aims.PointObject")
mPointObject.Y = -37.32
mPointObject.X = -122.7
```

See Also

Identify.asp
PointObject.X
Select_and Highlight.asp

Points.Add method

Description
Adds a point to a Points collection.

Syntax
Points.Add (Point as PointObject)

Argument
Point Point to add.

Returned Value
None

Example
```
Dim mPoints
Dim mPointObject
Set mPoints = Server.CreateObject("aims.Points")
Set mPointObject = Server.CreateObject("aims.PointObject")
mPoints.Add mPointObject
```

See Also
PointObject methods and properties

Points.Clear method

Description
Removes all points from the Points collection.

Syntax
Points.Clear()

Arguments
None

Returned Value
None

Example
```
Dim mPoints
Dim mPointObject
Set mPoints = Server.CreateObject("aims.Points")
Set mPointObject = Server.CreateObject("aims.PointObject")
mPoints.Add mPointObject
mPoints.Clear()
```

Points.Insert method

Description

Inserts a point at the specified position in a Points collection. The inserted point is placed before the point that was previously located at the specified position.

Syntax

Points.Insert(index as Long, Point as PointObject)

Arguments

Index	Point's position.
Point	Point to be inserted.

Returned Value

None

Example

```
Dim mPoints
Dim mPointObject
Set mPoints = Server.CreateObject("aims.Points")
Set mPointObject = Server.CreateObject("aims.PointObject")
mPoints.Insert 1, mPointObject
```

Points.Item method

Description

Returns a point by its index in a Points collection.

Syntax

Points.Item(index as Long)

Argument

Index	Point's position.

Returned Value

PointObject	Requested point.

Example

```
Dim mPoints
Dim mPointObject
Set mPoints = Server.CreateObject("aims.Points")
Set mPointObject = Server.CreateObject("aims.PointObject")
mPoints.Add (mPointObject)
Set mPointObject = mPoints.Item(1)
```

See Also

PointObject methods and properties

Points.Replace method

Description

Replaces a point in a Points collection.

Syntax

Points.Replace (index as Long, Point as PointObject)

Arguments

Index	Point's position.
Point	New point.

Returned Value

None

Example

```
Dim mPoints
Dim mPointObject
Set mPoints = Server.CreateObject("aims.Points")
Set mPointObject = Server.CreateObject("aims.PointObject")
Set newObject = Server.CreateObject("aims.PointObject")
mPoints.Add mPointObject
newObject.X = 5
mPoints.Replace 1, newObject
```

See Also

PointObject methods and properties

PolygonObject.Id property

Description

String identifier for a PolygonObject. The default value is Polygon. Write/Read.

Type

String

Example

```
dim mTestPt1
dim mTestPt2
dim mTestPt3
dim mTestPt4

set mTestPt1 = CreateObject("aims.PointObject")
set mTestPt2 = CreateObject("aims.PointObject")
set mTestPt3 = CreateObject("aims.PointObject")
set mTestPt4 = CreateObject("aims.PointObject")

mTestPt1.X = 100
mTestPt1.Y = 100
mTestPt2.X = 200
mTestPt2.Y = 200
mTestPt3.X = 100
mTestPt3.Y = 200
mTestPt4.X = 200
mTestPt4.Y = 100

dim mPoints
set mPoints = Server.CreateObject("aims.Points")

mPoints.Add mTestPt1
mPoints.Add mTestPt2
mPoints.Add mTestPt3
mPoints.Add mTestPt4

dim mPolygonObject
dim mSymbol
set mPolygonObject = Server.CreateObject("aims.PolygonObject")
mPolygonObject.Id = "Polygon1"
mPolygonObject.Name = "Two Triangles"
set mSymbol = mPolygonObject.Symbol
mPolygonObject.Parts.Add mPoints
mResult = AcetateLayer1.Add(mPolygonObject, 0)
```

PolygonObject.Name property

Description
String name for a PolygonObject. The default value is Polygon. Write/Read.

Type
String

Example
```
dim mPolygonObject
dim mSymbol
set mPolygonObject = Server.CreateObject("aims.PolygonObject")
mPolygonObject.Id = "Polygon1"
mPolygonObject.Name = "Two Triangles"
set mSymbol = mPolygonObject.Symbol

mResult = AcetateLayer1.Add(mPolygonObject, 0)
```

PolygonObject.Parts property

Description
The Parts object of a PolygonObject. The Parts object is a collection of Points objects that defines the geometry of PolygonObjects. Each item in a Parts object represents one polygon for a PolygonObject. Read only.

Type
Parts

Example
```
Dim mPolygonObject
Dim mParts
Set mPolygonObject = Server.CreateObject("aims.PolygonObject")
Set mParts = mPolygonObject.Parts
```

See Also
Parts methods and properties
Points methods and properties

PolygonObject.Symbol property

Description

Symbol object of the PolygonObject. The symbol object type is either SimplePolygonSymbol, RasterFillSymbol, or GradientFillSymbol. The default object is SimplePolygonSymbol. Write/Read.

Type

Symbol

Example

```
dim mTestPt1
dim mTestPt2
dim mTestPt3
dim mTestPt4

set mTestPt1 = CreateObject("aims.PointObject")
set mTestPt2 = CreateObject("aims.PointObject")
set mTestPt3 = CreateObject("aims.PointObject")
set mTestPt4 = CreateObject("aims.PointObject")

mTestPt1.X = 100
mTestPt1.Y = 100
mTestPt2.X = 200
mTestPt2.Y = 200
mTestPt3.X = 100
mTestPt3.Y = 200
mTestPt4.X = 200
mTestPt4.Y = 100

dim mPoints
set mPoints = Server.CreateObject("aims.Points")

mPoints.Add mTestPt1
mPoints.Add mTestPt2
mPoints.Add mTestPt3
mPoints.Add mTestPt4

dim mPolygonObject
dim mSymbol
set mPolygonObject = Server.CreateObject("aims.PolygonObject")
mPolygonObject.Id = "Polygon1"
mPolygonObject.Name = "Two Triangles"
set mSymbol = mPolygonObject.Symbol

mPolygonObject.Parts.Add mPoints

mResult = AcetateLayer1.Add(mPolygonObject, 0)
```

See Also

GradientFillSymbol methods and properties
RasterFillSymbol methods and properties
SimplePolygonSymbol methods and properties

RasterFillSymbol.Antialiasing property

Description

Boolean value indicating whether or not antialiasing will be used while rendering a RasterFillSymbol. Antialiasing allows lines to appear smooth by coloring adjacent pixels in such a way as to give the edge of a line a more gradual fade into the background. The default value is False. Write/Read.

Type

Boolean

Example

```
Dim mRasterFillSymbol
Set mRasterFillSymbol = Server.CreateObject("aims.RasterFillSymbol")
mRasterFillSymbol.Antialiasing = True
```

RasterFillSymbol.Boundary property

Description

Boolean value indicating whether or not a RasterFillSymbol's boundary will be drawn. The default value is False. Write/Read.

Type

Boolean

Example

```
Dim mRasterFillSymbol
Set mRasterFillSymbol = Server.CreateObject("aims.RasterFillSymbol")
mRasterFillSymbol.Boundary = True
```

RasterFillSymbol.Clone method

Description
Clones the RasterFillSymbol.

Syntax
RasterFillSymbol.Clone ()

Arguments
None

Returned Value
RasterFillSymbol Cloned symbol.

Example
```
dim mClone
dim mRasterFillSymbol
Set mRasterFillSymbol = Server.CreateObject("aims.RasterFillSymbol")
Set mClone = mRasterFillSymbol.Clone()
```

RasterFillSymbol.Image property

Description
Name of the image that will be used for the RasterFillSymbol. This could be a path to the image file. The ArcIMS Spatial Server reads this path to locate the image file, so the path will *not* be a URL but a normal file system path. Write/Read.

Type
String

Example
```
Dim mRasterFillSymbol
Set mRasterFillSymbol = Server.CreateObject("aims.RasterFillSymbol")
mRasterFillSymbol.Image = "C:\Temp\Picture.gif"
```

RasterFillSymbol.Overlap property

Description

Boolean value indicating whether or not labels will overlap a RasterFillSymbol. The default value is True. Write/Read.

Type

Boolean

Example

```
Dim mRasterFillSymbol
Set mRasterFillSymbol = Server.CreateObject("aims.RasterFillSymbol")
mRasterFillSymbol.Overlap = False
```

RasterFillSymbol.Transparency property

Description

Transparency coefficient for the RasterFillSymbol. Lower numbers will display the symbol with greater transparency. The default value is 1.0. Write/Read.

Type

Double

Example

```
Dim mRasterFillSymbol
Set mRasterFillSymbol = Server.CreateObject("aims.RasterFillSymbol")
mRasterFillSymbol.Transparency = 1.0
```

RasterFillSymbol.Url property

Description

URL to the image that will be used to fill a polygon feature. The client reads the URL to get the image for the RasterFillSymbol. Write/Read.

Type

String

Example

```
Dim mRasterFillSymbol
Set mRasterFillSymbol = Server.CreateObject("aims.RasterFillSymbol")
mRasterFillSymbol.URL = "http://www.esri.com/Picture.gif"
```

RasterMarkerSymbol.Antialiasing property

Description

Boolean value indicating whether or not antialiasing will be used while rendering a RasterMarkerSymbol. Antialiasing allows lines to appear smooth by coloring adjacent pixels in such a way as to give the edge of a line a more gradual fade into the background. The default value is False. Write/Read.

Type

Boolean

Example

```
Dim mRasterMarkerSymbol
Set mRasterMarkerSymbol = Server.CreateObject("aims.RasterMarkerSymbol")
mRasterMarkerSymbol.Antialiasing = True
```

RasterMarkerSymbol.Clone method

Description

Clones the RasterMarkerSymbol.

Syntax

RasterMarkerSymbol.Clone ()

Arguments

None

Returned Value

RasterMarkerSymbol Cloned symbol.

Example

```
dim mClone
dim mRasterMarkerSymbol
Set mRasterMarkerSymbol = Server.CreateObject("aims.RasterMarkerSymbol")
Set mClone = mRasterMarkerSymbol.Clone()
```

RasterMarkerSymbol.HotSpotX property

Description

X location where an image is placed in relation to a point. HotSpotX is the left coordinate. HotSpotX is always positive and is measured in pixels. Write/Read.

Type

Long

Example

```
Dim mRasterMarkerSymbol
Set mRasterMarkerSymbol = Server.CreateObject("aims.RasterMarkerSymbol")
mRasterMarkerSymbol.HotSpotX = 10
```

See Also

RasterMarkerSymbol.HotSpotY

RasterMarkerSymbol.HotSpotY property

Description

Y location where an image is placed in relation to a point. HotSpotY is the top coordinate. HotSpotY is always positive and is measured in pixels. Write/Read.

Type

Long

Example

```
Dim mRasterMarkerSymbol
Set mRasterMarkerSymbol = Server.CreateObject("aims.RasterMarkerSymbol")
mRasterMarkerSymbol.HotSpotY = 10
```

See Also

RasterMarkerSymbol.HotSpotX

RasterMarkerSymbol.Image property

Description

Name of the image that will be used to symbolize a point feature. This could be a path to the image file. The ArcIMS Spatial Server reads this path. Write/Read.

Type

String

Example

```
Dim mRasterMarkerSymbol
Set mRasterMarkerSymbol = Server.CreateObject("aims.RasterMarkerSymbol")
mRasterMarkerSymbol.Image = "C:\Temp\Picture.gif"
```

RasterMarkerSymbol.Overlap property

Description

Boolean value indicating whether or not labels will overlap a RasterMarkerSymbol. The default value is True. Write/Read.

Type

Boolean

Example

```
Dim mRasterMarkerSymbol
Set mRasterMarkerSymbol = Server.CreateObject("aims.RasterMarkerSymbol")
mRasterMarkerSymbol.Overlap = False
```

RasterMarkerSymbol.Shadow property

Description
Color constant for the shadow color of a RasterMarkerSymbol. The default value is imsBlack (0). Write/Read.

Type
Long

Example
```
Dim mRasterMarkerSymbol
Set mRasterMarkerSymbol = Server.CreateObject("aims.RasterMarkerSymbol")
mRasterMarkerSymbol.Shadow = imsColor.imsBlack
```

RasterMarkerSymbol.SizeX property

Description
X width of the RasterMarkerSymbol bit map. SizeX is always positive and is measured in pixels. Write/Read.

Type
Long

Example
```
Dim mRasterMarkerSymbol
Set mRasterMarkerSymbol = Server.CreateObject("aims.RasterMarkerSymbol")
mRasterMarkerSymbol.SizeX = 10
```

See Also
RasterMarkerSymbol.SizeY

RasterMarkerSymbol.SizeY property

Description
Y height of the RasterMarkerSymbol bit map. SizeY is always positive and is measured in pixels. Write/Read.

Type
Long

Example
```
Dim mRasterMarkerSymbol
Set mRasterMarkerSymbol = Server.CreateObject("aims.RasterMarkerSymbol")
mRasterMarkerSymbol.SizeY = 10
```

See Also
RasterMarkerSymbol.SizeX

RasterMarkerSymbol.Transparency property

Description

Transparency coefficient for the RasterMarkerSymbol. Lower numbers will display the symbol with greater transparency. The default value is 1.0. Write/Read.

Type

Double

Example

```
Dim mRasterMarkerSymbol
Set mRasterMarkerSymbol = Server.CreateObject("aims.RasterMarkerSymbol")
mRasterMarkerSymbol.Transparency = 1.0
```

RasterMarkerSymbol.Url property

Description

URL to the image that will be used to symbolize a point feature. The client reads the URL to get the image for the RasterMarkerSymbol. Write/Read.

Type

String

Example

```
Dim mRasterMarkerSymbol
Set mRasterMarkerSymbol = Server.CreateObject("aims.RasterMarkerSymbol")
mRasterMarkerSymbol.Url = "http://www.esri.com/Picture.gif"
```

RasterShieldSymbol.Antialiasing property

Description

Boolean value indicating whether or not antialiasing will be used while rendering the RasterShieldSymbol. Antialiasing allows lines to appear smooth by coloring adjacent pixels in such a way as to give the edge of a line a more gradual fade into the background. The default value is False. Write/Read.

Type

Boolean

Example

```
Dim mRasterShieldSymbol
Set mRasterShieldSymbol = Server.CreateObject("aims.RasterShieldSymbol")
mRasterShieldSymbol.Antialiasing = True
```

RasterShieldSymbol.Boundary property

Description

Boolean value indicating whether or not a boundary will be rendered for a RasterShieldSymbol. The default value is False. Write/Read.

Type

Boolean

Example

```
dim mRasterShieldSymbol
set mRasterShieldSymbol = Server.CreateObject("aims.RasterShieldSymbol")
mRasterShieldSymbol.Boundary = True
```

RasterShieldSymbol.Clone method

Description

Clones the RasterShieldSymbol.

Syntax

RasterShieldSymbol.Clone()

Arguments

None

Returned Value

RasterShieldSymbol Cloned symbol.

Example

```
dim mClone
dim mRasterShieldSymbol
Set mRasterShieldSymbol = Server.CreateObject("aims.RasterShieldSymbol")
Set mClone = mRasterShieldSymbol.Clone()
```

RasterShieldSymbol.Font property

Description

Name of the font used with the RasterShieldSymbol. The default value is default. Write/Read.

Type

String

Example

```
Dim mRasterShieldSymbol
Set mRasterShieldSymbol = Server.CreateObject("aims.RasterShieldSymbol")
mRasterShieldSymbol.Font = "Arial"
```

RasterShieldSymbol.FontColor property

Description

Font color constant for the font used with a RasterShieldSymbol. The default value is imsBlack (0). Write/Read.

Type

Long

Example

```
Dim mRasterShieldSymbol
Set mRasterShieldSymbol = Server.CreateObject("aims.RasterShieldSymbol")
mRasterShieldSymbol.Color = imsColor.imsRed
```

RasterShieldSymbol.FontSize property

Description

Font size for the font used with a RasterShieldSymbol. The default value is 12. Write/Read.

Type

Long

Example

```
Dim mRasterShieldSymbol
Set mRasterShieldSymbol = Server.CreateObject("aims.RasterShieldSymbol")
mRasterShieldSymbol.FontSize = 10
```

RasterShieldSymbol.FontStyle property

Description

Font style constant for the font used with a RasterShieldSymbol. The default value is imsRegular (0). The possible values are:

0 = imsRegular
1 = imsBold
2 = imsItalic
3 = imsUnderline
4 = imsOutline

Write/Read

Type

Long

Example

```
Dim mRasterShieldSymbol
Set mRasterShieldSymbol = Server.CreateObject("aims.RasterShieldSymbol")
mRasterShieldSymbol.FontStyle = imsFontStyle.imsBold
```

RasterShieldSymbol.Image property

Description

Name of the image that will be used for the RasterShieldSymbol. This could be a path to the image file. The ArcIMS Spatial Server reads this path. Write/Read.

Type

String

Example

```
Dim mRasterShieldSymbol
Set mRasterShieldSymbol = Server.CreateObject("aims.RasterShieldSymbol")
mRasterShieldSymbol.Image = "C:\Temp\Picture.gif"
```

RasterShieldSymbol.LabelMode property

Description

LabelMode constant that determines what labels are drawn in the RasterShieldSymbol. The default value is imsFull (0). The possible values are:

0 = imsFull
1 = imsNumericonly

“Full” means that the entire value of the label field will be displayed in the RasterShieldSymbol.
“Numericonly” means that only the numeric parts or the label field value will be displayed. Write/Read.

Type

Long

Example

```
Dim mRasterShieldSymbol
Set mRasterShieldSymbol = Server.CreateObject("aims.RasterShieldSymbol")
mRasterShieldSymbol.LabelMode = imsLabelMode.imsFull
```

See Also

imsLabelMode

RasterShieldSymbol.PrintMode property

Description

PrintMode constant that determines how RasterShieldSymbol labels will be rendered. The default value is imsNone (0). The possible values are:

0 = imsNone
1 = imsTitleCaps
2 = imsAllUpper
3 = imsAllLower

Write/Read

Type

Long

Example

```
Dim mRasterShieldSymbol
Set mRasterShieldSymbol = Server.CreateObject("aims.RasterShieldSymbol")
mRasterShieldSymbol.PrintMode = imsPrintMode.imsNone
```

See Also

imsPrintMode

RasterShieldSymbol.Shadow property

Description
Color constant for the shadow color of a RasterShieldSymbol. The default value is imsBlack (0). Write/Read.

Type
Long

Example
```
Dim mRasterShieldSymbol
Set mRasterShieldSymbol = Server.CreateObject("aims.RasterShieldSymbol")
mRasterShieldSymbol.Shadow = imsColor.imsBlack
```

RasterShieldSymbol.TextPosition property

Description
X,y location of the text on a map image. The default is the center of the image. Write/Read.

Type
String

Example
```
Dim mRasterShieldSymbol
Set mRasterShieldSymbol = Server.CreateObject("aims.RasterShieldSymbol")
mRasterShieldSymbol.TextPosition = "10,10"
```

RasterShieldSymbol.Transparency property

Description
Transparency coefficient for the RasterShieldSymbol. Lower numbers will display the symbol with greater transparency. The default value is 1.0. Write/Read.

Type
Double

Example
```
Dim mRasterShieldSymbol
Set mRasterShieldSymbol = Server.CreateObject("aims.RasterShieldSymbol")
mRasterShieldSymbol.Transparency = 1.0
```

RasterShieldSymbol.Url property

Description

URL to the image that will be used for a RasterShieldSymbol. The client reads the URL to get the image for the RasterShieldSymbol. Write/Read.

Type

String

Example

```
Dim mRasterShieldSymbol
Set mRasterShieldSymbol = Server.CreateObject("aims.RasterShieldSymbol")
mRasterShieldSymbol.URL = "http://www.esri.com/Picture.gif"
```

Recordset.BOF property

Description

Indicates that the current record position is before the first record in a Recordset object. BOF is true before the recordset is populated. A successful MoveFirst() call will set BOF to false. Read only.

Type

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mRecordset
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
Set mMap = Server.CreateObject("aims.Map")
mResult = mMap.InitMap(mArcIMSConnector, "World")
Set mRecordset = mMap.Layers.Item(1).Recordset
mRecordset.MoveLast()
While Not mRecordset.BOF
    mRecordset.MovePrevious()
Wend
```

See also

Recordset.EOF

Recordset.CacheSize property

Description

Number of records in a recordset that will be retrieved per a single request to the ArcIMS Spatial Server. The default is 10. Write/Read.

Type

Long

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mRecordset
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
Set mMap = Server.CreateObject("aims.Map")
mResult = mMap.InitMap(mArcIMSConnector, "World")
Set mRecordset = mMap.Layers.Item(1).Recordset
mRecordset.CacheSize = 15
```

See Also

MoveTo.asp

Recordset.Clone method

Description
Clones the recordset.

Syntax
Recordset.Clone()

Arguments
None

Returned Value
Recordset The clone of the recordset.

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mRecordset
Dim mClone
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
Set mMap = Server.CreateObject("aims.Map")
mResult = mMap.InitMap(mArcIMSConnector, "World")
Set mRecordset = mMap.Layers.Item(1).Recordset
Set mClone = mRecordset.Clone()
```

See Also
FeatureLayer.Clone

Recordset.Count property

Description
Number of records in a recordset. If the number of records is zero, then the count returns -1. Read only.

Type
Long

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mRecordset
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
Set mMap = Server.CreateObject("aims.Map")
mResult = mMap.InitMap(mArcIMSConnector, "World")
Set mRecordset = mMap.Layers.Item(1).Recordset
mCount = mRecordset.Count
```

Recordset.CurrentIndex property

Description

Returns the position/index of the current record in a recordset. The CurrentIndex is 1 after a successful MoveFirst call. Read only.

Type

Long

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mRecordset
Dim mCurrentIndex
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
Set mMap = Server.CreateObject("aims.Map")
mResult = mMap.InitMap(mArcIMSConnector, "World")
Set mRecordset = mMap.Layers.Item(1).Recordset
if mRecordset.MoveTo(3) then
   if (mRecordset.CurrentIndex <>3) then
      response.Write "Something must be wrong"
   end if
end if
```

See Also

MoveTo.asp
Recordset.MoveTo

Recordset.Envelope property

Description

Envelope for the current record in a recordset. The Envelope is the spatial extent of a record for a FeatureLayer feature. This property is not accessible until the recordset's MoveFirst or MoveLast methods have first been invoked. Read only.

Type

Envelope

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mRecordset
Dim mEnvelope
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
Set mMap = Server.CreateObject("aims.Map")
mResult = mMap.InitMap(mArcIMSConnector, "World")
Set mRecordset = mMap.Layers.Item(1).Recordset
if mRecordset.MoveFirst() then
Set mEnvelope = mRecordset.Envelope
end if
```

See Also

Envelope methods and properties

Recordset.EOF property

Description

Indicates that the current record position is after the last record in a Recordset object. EOF is true before the Recordset is populated. A successful MoveFirst() call will set BOF to false. Read only.

Type

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mRecordset
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
Set mMap = Server.CreateObject("aims.Map")
mResult = mMap.InitMap(mArcIMSConnector, "World")
Set mRecordset = mMap.Layers.Item(1).Recordset
While Not mRecordset.EOF
     mRecordset.MoveNext()
Wend
```

See Also

Recordset.BOF

Recordset.Fields property

Description

Fields object from the current recordset. The Fields object is not accessible until a MoveFirst() or MoveLast() method is invoked on the recordset. Read only.

Type

Fields

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mRecordset
Dim mFields
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
Set mMap = Server.CreateObject("aims.Map")
mResult = mMap.InitMap(mArcIMSConnector, "World")
Set mRecordset = mMap.Layers.Item(1).Recordset
if mRecordset.MoveFirst() then
     Set mFields = mRecordset.Fields
end if
```

See Also

Fields methods and properties
Identify.asp
Query_Attribute_Data.asp
Recordset.MoveFirst
Recordset.MoveLast
Recordset.MoveNext
Recordset.MovePrevious

Recordset.Filter property

Description

Filter object for the recordset. The filter of a recordset from a FeatureLayer is independent of the FeatureLayer's filter object. Write/Read.

Type

Filter

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mFilter
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
Set mMap = Server.CreateObject("aims.Map")
mResult = mMap.InitMap(mArcIMSConnector, "World")
Set mFilter = mMap.Layers.Item(1).Recordset.Filter
```

See Also

FeatureLayer.Filter
Filter methods and properties
Identify.asp
Query_Attribute_Data.asp

Recordset.MoveFirst method

Description

Sets the internal pointer to the first record in a recordset and returns a Boolean value indicating the success of the method. If the recordset is empty, MoveFirst() will return False.

Syntax

Recordset.MoveFirst()

Arguments

None

Returned Value

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mRecordset
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
Set mMap = Server.CreateObject("aims.Map")
mResult = mMap.InitMap(mArcIMSConnector, "World")
Set mRecordset = mMap.Layers.Item(1).Recordset
mResult = mRecordset.MoveFirst()
```

See Also

Identify.asp
Query_Attribute_Data.asp
Recordset.MoveLast
Recordset.MoveNext
Recordset.MovePrevious

Recordset.MoveLast method

Description

Sets the internal pointer to the last record in a recordset and returns a Boolean value indicating the success of the method. If the recordset is empty, MoveLast() will return False.

Syntax

Recordset.MoveLast()

Arguments

None

Returned Value

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mRecordset
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
Set mMap = Server.CreateObject("aims.Map")
mResult = mMap.InitMap(mArcIMSConnector, "World")
Set mRecordset = mMap.Layers.Item(1).Recordset
mResult = mRecordset.MoveLast()
```

See Also

Recordset.MoveFirst
Recordset.MoveNext
Recordset.MovePrevious

Recordset.MoveNext method

Description

Sets the internal pointer to the next record in a recordset and returns a Boolean value indicating the success of the method. If the recordset pointer is at the last record, MoveNext() will return False.

Syntax

Recordset.MoveNext()

Arguments

None

Returned Value

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mRecordset
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
Set mMap = Server.CreateObject("aims.Map")
mResult = mMap.Init(mArcIMSConnector, "World")
Set mRecordset = mMap.Layers.Item(1).Recordset
if mRecordset.MoveFirst() then
   Do
   ...
  Loop While mRecordset.MoveNext()
end if
```

See Also

Query_Attribute_Data.asp
Recordset.MoveFirst
Recordset.MoveLast
Recordset.MovePrevious

Recordset.MovePrevious method

Description

Sets the internal pointer to the previous record in a recordset and returns a Boolean value indicating the success of the method. If the recordset pointer is at the first record, MovePrevious() will return False.

Syntax

Recordset.MovePrevious()

Arguments

None

Returned Value

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mRecordset
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
Set mMap = Server.CreateObject("aims.Map")
mResult = mMap.InitMap(mArcIMSConnector, "World")
Set mRecordset = mMap.Layers.Item(1).Recordset
mResult = mRecordset.MoveLast()
mResult = mRecordset.MovePrevious()
```

See Also

Recordset.MoveFirst
Recordset.MoveLast
Recordset.MoveNext

Recordset.MoveTo method

Description

Sets the internal pointer to the specified position and returns a Boolean value indicating the success of the method. If the specified position is less than one or larger than the count of the Recordset, MoveTo returns False. If the recordset is empty, MoveTo returns False.

Syntax

Recordset.MoveTo(index)

Arguments

index Long A position in the Recordset object.

Returned Value

Boolean

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mRecordset
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
Set mMap = Server.CreateObject("aims.Map")
mResult = mMap.InitMap(mArcIMSConnector, "World")
Set mRecordset = mMap.Layers.Item(1).Recordset
mResult = mRecordset.MoveTo(1)  ' same as MoveFirst
```

See Also

MoveTo.asp
Recordset.MoveFirst
Recordset.MoveLast
Recordset.MoveNext
Recordset.MovePrevious
Recordset.CurrentIndex

Recordset.TableDesc property

Description

TableDesc object that contains a description of the fields in a recordset. Read only.

Type

TableDesc

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mTableDesc
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
Set mMap = Server.CreateObject("aims.Map")
mResult = mMap.InitMap(mArcIMSConnector, "World")
Set mTableDesc = mMap.Layers.Item(1).Recordset.TableDesc
```

See Also

TableDesc methods and properties
Query_Attribute_Data.asp

ScaleBarObject.Antialiasing property

Description

Turns antialiasing on/off. Antialiasing allows lines to appear smooth by coloring adjacent pixels in such a way as to give the edge of a line a more gradual fade into the background. Write/Read.

Type

Boolean

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

ScaleBarObject.BarColor property

Description

Color constant for the ScaleBarObject color. The default value is 0. Write/Read.

Type

Long

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

ScaleBarObject.BarTransparency property

Description

ScaleBarObject's transparency coefficient. A lower number indicates greater transparency for the ScaleBarObject. The default value is 1. Write/Read.

Type

Double

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mScaleBar.BarTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

ScaleBarObject.BarWidth property

Description

ScaleBarObject width in pixels. Write/Read.

Type

Long

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

ScaleBarObject.Distance property

Description

Distance of the ScaleBarObject in map units. The default value is 0. This value is the calculated distance of the setting map unit from the value of the Scalebar.ScreenLength. Write/Read.

Type

Double

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mScaleBar.Distance = 100
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

ScaleBarObject.Font property

Description

Font name string for the text labels used in the ScaleBarObject. The default value is Arial. Write/Read.

Type

String

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

ScaleBarObject.FontColor property

Description

Color constant for the font of the text labels that are used for the ScaleBarObject. The default value is imsBlack (0). Write/Read.

Type

Long

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

See Also

imsColor

ScaleBarObject.FontSize property

Description

Font size for text labels used with the ScaleBarObject. Write/Read.

Type

Long

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mScaleBar.BarTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

ScaleBarObject.FontStyle property

Description

Font style constant for the text used with the ScaleBarObject. The default value is imsRegular (0). Possible values are:

0 = imsRegular
1 = imsBold
2 = imsItalic
3 = imsUnderline
4 = imsOutline

Write/Read.

Type

Long

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mScaleBar.BarTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

See Also

imsFontStyle

ScaleBarObject.Id property

Description

String identifier for a ScaleBarObject. The default value is ScaleBar. Write/Read.

Type

String

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mScaleBar.BarTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

ScaleBarObject.MapUnits property

Description

Constant that indicates the units of the ScaleBarObject. The default value is imsDecimalDegrees (0).
Available values are:

0 = imsDecimalDegrees
1 = imsMiles
2 = imsFeet
3 = imsKilometers
4 = imsMeters
5 = imsInches
6 = imsCentimeters

Write/Read.

Type

Long

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mScaleBar.BarTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

See Also

imsMapUnits

ScaleBarObject.Mode property

Description

ScaleBar Mode. Used when the map units are in decimal degrees. When the mode is geodesic, the Image Server takes into account the position on the globe when calculating the size of the scalebar symbol. When the mode is cartesian, the Image Server uses the same calculation for the scalebar for all points on the globe. The calculation is made at the equator. Write/Read.

Type

Long

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0 , degrees
mScalebar.ScaleUnits = 0 , kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mScaleBar.BarTransparency = 1.0
mScaleBar.Mode = 0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

ScaleBarObject.Name property

Description

Name of the ScaleBarObject. The default value is ScaleBar. Write/Read.

Type

String

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mScaleBar.BarTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

ScaleBarObject.Overlap property

Description

Boolean value indicating whether or not labels should be allowed to overlap the ScaleBarObject. The default value is False. Write/Read.

Type

Boolean

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mScaleBar.BarTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

ScaleBarObject.Precision property

Description

Number of decimal places displayed for the ScaleBarObject units. Write/Read.

Type

Long

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mScaleBar.BarTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

ScaleBarObject.ScaleUnits property

Description

Constant indicating the display units of the ScaleBarObject. The default value is imsDecimalDegrees (0). The available values are:

0 = imsDecimalDegrees
1 = imsMiles
2 = imsFeet
3 = imsKilometers
4 = imsMeters
5 = imsInches
6 = imsCentimeters

Write/Read.

Type

Long

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mScaleBar.BarTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

See Also

imsMapUnits

ScaleBarObject.ScreenLength property

Description

ScaleBarObject length in screen pixels. Write/Read.

Type

Double

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mScaleBar.BarTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

ScaleBarObject.TextTransparency property

Description

Coefficient of the text transparency for the ScaleBarObject's text. The lower the number, the greater the transparency. The default value is 1.0. Write/Read.

Type

Double

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mScaleBar.BarTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

ScaleBarObject.X property

Description

X coordinate of the ScaleBarObject location. The default value is 0. Write/Read.

Type

Double

Example

```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.FontName = "Arial"
mScaleBar.FontColor = imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = acBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

See Also

ScaleBarObject.Y

ScaleBarObject.Y property

Description
Y coordinate of the ScaleBarObject location. The default value is 0. Write/Read.

Type
Double

Example
```
dim mScalebar
set mScalebar = CreateObject("aims.ScaleBarObject")
mScalebar.x = 25
mScalebar.y = 25
mScalebar.BarWidth = 10.0
mScaleBar.Font = "Arial"
mScaleBar.FontColor = imsColor.imsBlack
mScalebar.FontSize = 12
mScaleBarObject.FontStyle = imsFontStyle.imsRegular
mScalebar.BarColor = imsColor.imsBlack
mScalebar.MapUnits = 0  ' degrees
mScalebar.ScaleUnits = 0  ' kilometers
mScalebar.ScreenLength = 50
mScalebar.Antialiasing = true
mScaleBar.Overlap = False
mScaleBar.Id = "ScaleBar:1"
mScaleBar.Name = "MyScaleBar"
mScaleBar.Precision = 2
mScaleBar.TextTransparency = 1.0
mResult = Map.Layers.Add(AcetateLayer1)
mResult = AcetateLayer1.Add(mScaleBar, 0)
```

See Also
ScaleBarObject.X

ScaleDependentRenderer.Clone method

Description
Clones the ScaleDependentRenderer.

Syntax
ScaleDependentRenderer.Clone()

Arguments
None

Returned Value
ScaleDependentRenderer Cloned renderer.

Example
```
dim mClone
dim mScaleDependentRenderer
set mScaleDependentRenderer = Server.CreateObject("aims.ScaleDependentRenderer")
set mClone = mScaleDependentRenderer.Clone()
mClone.Lower = 0.1
mClone.Upper = 0.01
set mRenderer = Server.CreateObject("aims.SimpleRenderer")
mClone.Renderer = mRenderer
```

ScaleDependentRenderer.Lower property

Description
Lower relative scale threshold of a ScaleDependentRenderer. Write/Read.

Type
Double

Example
```
dim mClone
dim mScaleDependentRenderer
set mScaleDependentRenderer = Server.CreateObject("aims.ScaleDependentRenderer")
set mClone = mScaleDependentRenderer.Clone()
mClone.Lower = 0.1
mClone.Upper = 0.01
set mRenderer = Server.CreateObject("aims.SimpleRenderer")
mClone.Renderer = mRenderer
```

See Also
ScaleDependentRenderer.Upper

ScaleDependentRenderer.Renderer property

Description

Renderer object that is to be displayed by the ScaleDependentRenderer. The Renderer will be under the constraints of the ScaleDependentRenderer properties. Write/Read.

Type

Object

Example

```
dim mClone
dim mScaleDependentRenderer
set mScaleDependentRenderer = Server.CreateObject("aims.ScaleDependentRenderer")
set mClone = mScaleDependentRenderer.Clone()
mClone.Lower = 0.1
mClone.Upper = 0.01
set mRenderer = Server.CreateObject("aims.SimpleRenderer")
mClone.Renderer = mRenderer
```

ScaleDependentRenderer.Upper property

Description

Upper relative scale threshold for the ScaleDependentRenderer. The default value is 0. Write/Read.

Type

Double

Example

```
dim mClone
dim mScaleDependentRenderer
set mScaleDependentRenderer = Server.CreateObject("aims.ScaleDependentRenderer")
set mClone = mScaleDependentRenderer.Clone()
mClone.Lower = 0.1
mClone.Upper = 0.01
set mRenderer = Server.CreateObject("aims.SimpleRenderer")
mClone.Renderer = mRenderer
```

See Also

ScaleDependentRenderer.Lower

SDEWorkspace.DataBase property

Description

ArcSDE database name for an SDEWorkspace. Write/Read.

Type

String

Example

```
dim mSDEWorkspace
dim lay
dim mConn
dim mMap
set mConn = server.CreateObject("aims.ArcIMSConnector")
mConn.ServerName = "husker"
mConn.ServerPort = 5300
set mMap = server.CreateObject("aims.Map")
mMap.InitMap mConn,"World"
Set mSDEWorkspace = Server.CreateObject("aims.SDEWorkspace")
MSDEWorkspace.DataBase = "SDE1"
mSDEWorkspace.User = "sde"
mSDEWorkspace.Password = "sde"
mSDEWorkspace.Server = "brie"
mSDEWorkspace.Instance = "brie_sde"
mSDEWorkspace.name = "SDE_AWS_CONT"
mSDEWorkspace.FeatureClass = imsFeatureClass.imsPolygon
set Lay = mMap.Layers.Create(mSDEWorkSpace)
mMap.Refresh
```

SDEWorkspace.FeatureClass property

Description

FeatureClass type constant of the layer to be specified in the SDEWorkspace. The default value is 0. The possible values are:

1 = imsPoint
2 = imsLine
3 = imsPolygon

Write/Read.

Type

Long

Example

```
dim mSDEWorkspace
dim lay
dim mConn
dim mMap
set mConn = server.CreateObject("aims.ArcIMSConnector")
mConn.ServerName = "husker"
mConn.ServerPort = 5300
set mMap = server.CreateObject("aims.Map")
mMap.InitMap mConn,"World"
Set mSDEWorkspace = Server.CreateObject("aims.SDEWorkspace")
mSDEWorkspace.User = "mwd"
mSDEWorkspace.Password = "mwd"
mSDEWorkspace.Server = "brie"
mSDEWorkspace.FeatureClass = imsFeatureClass.imsPoint
mSDEWorkspace.Name = "mwd.facil.gid"
set lay = mMap.Layers.Create(mSDEWorkspace)
```

SDEWorkspace.Geoindexer property

Description

Name of the directory where the geocoding index will be built for the SDEWorkspace layer. Write/Read.

Type

String

Example

```
dim mSDEWorkspace
dim lay
dim mConn
dim mMap
set mConn = server.CreateObject("aims.ArcIMSConnector")
mConn.ServerName = "husker"
mConn.ServerPort = 5300
set mMap = server.CreateObject("aims.Map")
mMap.InitMap mConn,"World"
Set mSDEWorkspace = Server.CreateObject("aims.SDEWorkspace")
mSDEWorkspace.User = "mwd"
mSDEWorkspace.Password = "mwd"
mSDEWorkspace.Server = "brie"
mSDEWorkspace.FeatureClass = imsFeatureClass.imsLine
mSDEWorkspace.Name = "mwd.streets.gid"
mSDEWorkspace.Geoindexer = "C:\Temp\Indexer"
```

SDEWorkspace.Id property

Description

String identifier for the SDEWorkspace. The default value begins with sde_ws- followed by a random number (for example, sde_ws-29816). Write/Read.

Type

String

Example

```
dim mSDEWorkspace
dim lay
dim mConn
dim mMap
set mConn = server.CreateObject("aims.ArcIMSConnector")
mConn.ServerName = "husker"
mConn.ServerPort = 5300
set mMap = server.CreateObject("aims.Map")
mMap.InitMap mConn,"World"
Set mSDEWorkspace = Server.CreateObject("aims.SDEWorkspace")
mSDEWorkspace.User = "mwd"
mSDEWorkspace.Password = "mwd"
mSDEWorkspace.Server = "brie"
mSDEWorkspace.FeatureClass = imsFeatureClass.imsLine
mSDEWorkspace.Name = "mwd.streets.gid"
mSDEWorkspace.Id = "SDEWorkspace:3"
```

SDEWorkspace.Instance property

Description

String indicating an ArcSDE instance name for the SDEWorkspace. Write/Read.

Type

String

Example

```
dim mSDEWorkspace
dim lay
dim mConn
dim mMap
set mConn = server.CreateObject("aims.ArcIMSConnector")
mConn.ServerName = "husker"
mConn.ServerPort = 5300
set mMap = server.CreateObject("aims.Map")
mMap.InitMap mConn,"World"
Set mSDEWorkspace = Server.CreateObject("aims.SDEWorkspace")
mSDEWorkspace.User = "mwd"
mSDEWorkspace.Password = "mwd"
mSDEWorkspace.Server = "brie"
mSDEWorkspace.FeatureClass = imsFeatureClass.imsLine
mSDEWorkspace.Name = "mwd.streets.gid"
mSDEWorkspace.Instance = "port:5151"
set Lay = mMap.Layers.Create(mSDEWorkSpace)
mMap.Refresh
```

SDEWorkspace.Name property

Description

Name of a layer to be specified in an SDEWorkspace. Write/Read.

Type

String

Example

```
dim mSDEWorkspace
dim lay
dim mConn
dim mMap
set mConn = server.CreateObject("aims.ArcIMSConnector")
mConn.ServerName = "husker"
mConn.ServerPort = 5300
set mMap = server.CreateObject("aims.Map")
mMap.InitMap mConn,"World"
Set mSDEWorkspace = Server.CreateObject("aims.SDEWorkspace")
mSDEWorkspace.User = "mwd"
mSDEWorkspace.Password = "mwd"
mSDEWorkspace.Server = "brie"
mSDEWorkspace.FeatureClass = imsFeatureClass.imsLine
mSDEWorkspace.Name = "Country"
```

SDEWorkspace.Password property

Description
ArcSDE password string of a user in an SDEWorkspace. Write/Read.

Type
String

Example
```
dim mSDEWorkspace
dim lay
dim mConn
dim mMap
set mConn = server.CreateObject("aims.ArcIMSConnector")
mConn.ServerName = "husker"
mConn.ServerPort = 5300
set mMap = server.CreateObject("aims.Map")
mMap.InitMap mConn,"World"
Set mSDEWorkspace = Server.CreateObject("aims.SDEWorkspace")
mSDEWorkspace.User = "mwd"
mSDEWorkspace.Password = "mwd"
mSDEWorkspace.Server = "brie"
mSDEWorkspace.FeatureClass = imsFeatureClass.imsLine
mSDEWorkspace.Name = "Country"

set Lay = mMap.Layers.Create(mSDEWorkSpace)
mMap.Refresh
```

SDEWorkspace.Server property

Description
Name of an ArcSDE server that the SDEWorkspace object will reference. Write/Read.

Type
String

Example
```
dim mSDEWorkspace
dim lay
dim mConn
dim mMap
set mConn = server.CreateObject("aims.ArcIMSConnector")
mConn.ServerName = "husker"
mConn.ServerPort = 5300
set mMap = server.CreateObject("aims.Map")
mMap.InitMap mConn,"World"
Set mSDEWorkspace = Server.CreateObject("aims.SDEWorkspace")
mSDEWorkspace.User = "mwd"
mSDEWorkspace.Password = "mwd"
mSDEWorkspace.Server = "brie"
mSDEWorkspace.FeatureClass = imsFeatureClass.imsLine
mSDEWorkspace.Name = "Country"
set Lay = mMap.Layers.Create(mSDEWorkSpace)
mMap.Refresh
```

SDEWorkspace.User property

Description

ArcSDE username for an SDEWorkspace object. Write/Read.

Type

String

Example

```
dim mSDEWorkspace
dim lay
dim mConn
dim mMap
set mConn = server.CreateObject("aims.ArcIMSConnector")
mConn.ServerName = "husker"
mConn.ServerPort = 5300
set mMap = server.CreateObject("aims.Map")
mMap.InitMap mConn,"World"
Set mSDEWorkspace = Server.CreateObject("aims.SDEWorkspace")
mSDEWorkspace.User = "mwd"
mSDEWorkspace.Password = "mwd"
mSDEWorkspace.Server = "brie"
mSDEWorkspace.FeatureClass = imsFeatureClass.imsLine
mSDEWorkspace.Name = "Country"
set Lay = mMap.Layers.Create(mSDEWorkSpace)
mMap.Refresh
```

ShapeWorkspace.Directory property

Description

Pathname of a directory containing shapefile data. Write/Read.

Type

String

Example

```
dim mShapeWorkspace
dim lay
dim mConn
dim mMap
set mConn = server.CreateObject("aims.ArcIMSConnector")
mConn.ServerName = "husker"
mConn.ServerPort = 5300
set mMap = server.CreateObject("aims.Map")
set mShapeWorkspace = Server.CreateObject("aims.ShapeWorkspace")
mShapeWorkspace.Directory = "D:\EsriData\Shapes"
mShapeWorkspace.FeatureClass = imsFeatureClass.imsPolygon
mShapeWorkspace.Name = "Countries"
set Lay = mMap.Layers.Create(mShapeWorkSpace)
mMap.Refresh
```

ShapeWorkspace.FeatureClass property

Description

Type of feature class for a layer to be specified in a ShapeWorkspace. The possible values are:

1 = imsPoint
2 = imsLine
3 = imsPolygon

Write/Read.

Type

Long

Example

```
dim mShapeWorkspace
dim lay
dim mConn
dim mMap
set mConn = server.CreateObject("aims.ArcIMSConnector")
mConn.ServerName = "husker"
mConn.ServerPort = 5300
set mMap = server.CreateObject("aims.Map")
set mShapeWorkspace = Server.CreateObject("aims.ShapeWorkspace")
mShapeWorkspace.Directory = "D:\EsriData\Shapes"
mShapeWorkspace.FeatureClass = imsFeatureClass.imsPolygon
mShapeWorkspace.Name = "Countries"
set Lay = mMap.Layers.Create(mShapeWorkSpace)
mMap.Refresh
```

See Also

imsFeatureClass

ShapeWorkspace.Id property

Description

String identifier for a ShapeWorkspace. The default value begins with shp_ws- and is followed by a random number (for example, shp_ws-76072). Write/Read.

Type

String

Example

```
dim mShapeWorkspace
dim lay
dim mConn
dim mMap
set mConn = server.CreateObject("aims.ArcIMSConnector")
mConn.ServerName = "husker"
mConn.ServerPort = 5300
set mMap = server.CreateObject("aims.Map")
set mShapeWorkspace = Server.CreateObject("aims.ShapeWorkspace")
mShapeWorkspace.Directory = "D:\EsriData\Shapes"
mShapeWorkspace.FeatureClass = imsFeatureClass.imsPolygon
mShapeWorkspace.Name = "Countries"
mShapeWorkspace.Id = "ID1"
set Lay = mMap.Layers.Create(mShapeWorkSpace)
mMap.Refresh
```

ShapeWorkspace.Name property

Description

Name of the layer to be specified in the ShapeWorkspace. Write/Read.

Type

String

Example

```
dim mShapeWorkspace
dim lay
dim mConn
dim mMap
set mConn = server.CreateObject("aims.ArcIMSConnector")
mConn.ServerName = "husker"
mConn.ServerPort = 5300
set mMap = server.CreateObject("aims.Map")
set mShapeWorkspace = Server.CreateObject("aims.ShapeWorkspace")
mShapeWorkspace.Directory = "D:\EsriData\Shapes"
mShapeWorkspace.FeatureClass = imsFeatureClass.imsPolygon
mShapeWorkspace.Name = "Countries"
set Lay = mMap.Layers.Create(mShapeWorkSpace)
mMap.Refresh
```

ShieldSymbol.Clone method

Description
Clones the ShieldSymbol.

Syntax
ShieldSymbol.Clone()

Arguments
None

Returned Value
ShieldSymbol Cloned symbol.

Example
```
dim mClone
dim mShieldSymbol
Set mShieldSymbol = Server.CreateObject("aims.ShieldSymbol")
Set mClone = mShieldSymbol.Clone()
```

ShieldSymbol.Font property

Description
Name of the font used with the ShieldSymbol. The default value is default. Write/Read.

Type
String

Example
```
Dim mShieldSymbol
Set mShieldSymbol = Server.CreateObject("aims.ShieldSymbol")
mShieldSymbol.Font = "Arial"
```

ShieldSymbol.FontColor property

Description
Font color constant for the font used with a ShieldSymbol. The default value is imsBlack (0). Write/Read.

Type
Long

Example
```
Dim mShieldSymbol
Set mShieldSymbol = Server.CreateObject("aims.ShieldSymbol")
mShieldSymbol.FontColor = imsColor.imsRed
```

ShieldSymbol.FontSize property

Description

Font size for the font used with a ShieldSymbol. The default value is 12. Write/Read.

Type

Long

Example

```
Dim mShieldSymbol
Set mShieldSymbol = Server.CreateObject("aims.ShieldSymbol")
mShieldSymbol.FontSize = 10
```

ShieldSymbol.LabelMode property

Description

LabelMode constant that determines what labels are drawn in the ShieldSymbol. The default value is imsFull (0). The possible values are:

0 = imsFull
1 = imsNumericonly

“Full” means that the entire value of the label field will be displayed in the ShieldSymbol. “Numericonly” means that only the numeric parts of the label field value will be displayed.

Write/Read.

Type

Long

Example

```
Dim mShieldSymbol
Set mShieldSymbol = Server.CreateObject("aims.ShieldSymbol")
mShieldSymbol.LabelMode = imsLabelMode.imsFull
```

See Also

imsLabelMode

ShieldSymbol.MinSize property

Description

Minimum ShieldSymbol size. The size determines the minimum size in characters. By default, shields expand to the length of the text. Write/Read.

Type

Long

Example

```
Dim mShieldSymbol
Set mShieldSymbol = Server.CreateObject("aims.ShieldSymbol")
mShieldSymbol.MinSize = 10
```

ShieldSymbol.Shadow property

Description

Color constant for the shadow color of a ShieldSymbol. The default value is imsBlack (0). Write/Read.

Type

Long

Example

```
Dim mShieldSymbol
Set mShieldSymbol = Server.CreateObject("aims.ShieldSymbol")
mShieldSymbol.Shadow = imsColor.imsBlack
```

ShieldSymbol.ShieldType property

Description

Shield type constant for a ShieldSymbol. The default value is imsInterstate (0). The possible values are:

0 = imsInterstate
1 = imsUSRoad
2 = imsRect
3 = imsOval
4 = imsMexican

Write/Read.

Type

Long

Example

```
Dim mShieldSymbol
Set mShieldSymbol = Server.CreateObject("aims.ShieldSymbol")
mShieldSymbol.ShieldType = imsShieldType.imsRect
```

See Also

imsShieldType

SimpleLabelRenderer.Clone method

Description
Clones the SimpleLabelRenderer.

Syntax
SimpleLabelRenderer.Clone()

Arguments
None

Returned Value
SimpleLabelRenderer Cloned renderer.

Example
```
dim mClone
dim mSimpleLabelRenderer
set mSimpleLabelRenderer = Server.CreateObject("aims.SimpleLabelRenderer")
set mClone = mSimpleLabelRenderer.Clone()
mSimpleLabelRenderer.Field = "CNTRY_NAME"
mSimpleLabelRenderer.FWeight = imsWeight.imsNo_weight
mSimpleLabelRenderer.HMLabels = imsHMLabels.imsOne_label_per_name
mSimpleLabelRenderer.LabelPriorities = "1,1,1,1,2,2,2,2"
mSimpleLabelRenderer.LLPosition = imsLLPosition.imsPlaceNone
mSimpleLabelRenderer.RotationalAngles = 30
mSimpleLabelRenderer.Symbol = mTextSymbol
mMap.Refresh()
```

SimpleLabelRenderer.Field property

Description
Field name that contains the values for labeling features. Write/Read.

Type
String

Example
```
dim mClone
dim mSimpleLabelRenderer
set mSimpleLabelRenderer = Server.CreateObject("aims.SimpleLabelRenderer")
set mClone = mSimpleLabelRenderer.Clone()
mSimpleLabelRenderer.Field = "CNTRY_NAME"
mSimpleLabelRenderer.FWeight = imsWeight.imsNo_weight
mSimpleLabelRenderer.HMLabels = imsHMLabels.imsOne_label_per_name
mSimpleLabelRenderer.LabelPriorities = "1,1,1,1,2,2,2,2"
mSimpleLabelRenderer.LLPosition = imsLLPosition.imsPlaceNone
mSimpleLabelRenderer.RotationalAngles = 30
mSimpleLabelRenderer.Symbol = mTextSymbol
mMap.Refresh()
```

SimpleLabelRenderer.FWeight property

Description

Prioritizes the importance of features. The feature weight determines how important the feature labeled is for the label placement algorithm. If imsNo_weight is specified, then the feature has no importance and can be labeled over. If imsHigh_weight is specified, then the feature has high importance and cannot be labeled over. The default value is imsNo_weight.

The possible values are:

0 = imsNo_weight
1 = imsMed_weight
2 = imsHigh_weight

Write/Read.

Type

Long

Example

```
dim mClone
dim mSimpleLabelRenderer
set mSimpleLabelRenderer = Server.CreateObject("aims.SimpleLabelRenderer")
set mClone = mSimpleLabelRenderer.Clone()
mSimpleLabelRenderer.Field = "CNTRY_NAME"
mSimpleLabelRenderer.FWeight = imsWeight.imsNo_weight
mSimpleLabelRenderer.HMLabels = imsHMLabels.imsOne_label_per_name
mSimpleLabelRenderer.LabelPriorities = "1,1,1,1,2,2,2,2"
mSimpleLabelRenderer.LLPosition = imsLLPosition.imsPlaceNone
mSimpleLabelRenderer.RotationalAngles = 30
mSimpleLabelRenderer.Symbol = mTextSymbol
mMap.Refresh()
```

SimpleLabelRenderer.HMLabels property

Description

Constant indicating how many labels are to be drawn while labeling a feature. The default value is imsOne_label_per_name (0). The possible values are:

0 = imsOne_label_per_name
1 = imsOne_label_per_shape
2 = imsOne_label_per_part

Write/Read

Type

Long

Example

```
dim mClone
dim mSimpleLabelRenderer
set mSimpleLabelRenderer = Server.CreateObject("aims.SimpleLabelRenderer")
set mClone = mSimpleLabelRenderer.Clone()
mSimpleLabelRenderer.Field = "CNTRY_NAME"
mSimpleLabelRenderer.FWeight = imsWeight.imsNo_weight
mSimpleLabelRenderer.HMLabels = imsHMLabels.imsOne_label_per_name
mSimpleLabelRenderer.LabelPriorities = "1,1,1,1,2,2,2,2"
mSimpleLabelRenderer.LLPosition = imsLLPosition.imsPlaceNone
mSimpleLabelRenderer.RotationalAngles = 30
mSimpleLabelRenderer.Symbol = mTextSymbol
mMap.Refresh()
```

See Also

imsHMLabels

SimpleLabelRenderer.LabelBufferRatio property

Description

Value used to set a buffer around the label. When this is set, no labels overlap within the buffer range. The ratio is the fraction of the height or the width of the label rectangle (whichever is smaller) compared to the width of the buffer. A ratio of 0.0 means no buffer. A ratio of 1.0 means that the buffer is twice the size of the label (the label width equals the buffer width). A negative ratio causes the buffer to be smaller than the label. This can be used to allow labels to overlap. Write/Read.

Type

Double

Example

```
dim mClone
dim mSimpleLabelRenderer
set mSimpleLabelRenderer = Server.CreateObject("aims.SimpleLabelRenderer")
set mClone = mSimpleLabelRenderer.Clone()
mSimpleLabelRenderer.Field = "CNTRY_NAME"
mSimpleLabelRenderer.FWeight = imsWeight.imsNo_weight
mSimpleLabelRenderer.HMLabels = imsHMLabels.imsOne_label_per_name
mSimpleLabelRenderer.LabelPriorities = "1,1,1,1,2,2,2,2"
mSimpleLabelRenderer.LLPosition = imsLLPosition.imsPlaceNone
mSimpleLabelRenderer.RotationalAngles = 30
mSimpleLabelRenderer.LabelBufferRatio = 1.0
mSimpleLabelRenderer.Symbol = mTextSymbol
mMap.Refresh()
```

SimpleLabelRenderer.LabelPriorities property

Description

String that determines where to place a label around a point. There are eight positions around a point. The LabelPriorities string is a comma-separated list of priorities for the label location at each of the eight positions: 1 being the highest, 8 being the lowest. For example, 1,1,0,0,2,3,4,4 means the following: try to place the label at locations 1 and 2. Do not place the label at locations 3 or 4. If the label can't be placed at 1 or 2, try locations 5, then 6, then 7 or 8. The default string is 2,2,1,4,5,3,2,4. Write/Read.

Type

String

Example

```
dim mClone
dim mSimpleLabelRenderer
set mSimpleLabelRenderer = Server.CreateObject("aims.SimpleLabelRenderer")
set mClone = mSimpleLabelRenderer.Clone()
mSimpleLabelRenderer.Field = "CNTRY_NAME"
mSimpleLabelRenderer.FWeight = imsWeight.imsNo_weight
mSimpleLabelRenderer.HMLabels = imsHMLabels.imsOne_label_per_name
mSimpleLabelRenderer.LabelPriorities = "1,1,1,1,2,2,2,2"
mSimpleLabelRenderer.LLPosition = imsLLPosition.imsPlaceNone
mSimpleLabelRenderer.RotationalAngles = 30
mSimpleLabelRenderer.Symbol = mTextSymbol
mMap.Refresh()
```

SimpleLabelRenderer.LLPosition property

Description

Constant determining where to place a label on a line. The default value is imsPlaceAbove. Write/Read.

Type

Long

Example

```
dim mClone
dim mSimpleLabelRenderer
set mSimpleLabelRenderer = Server.CreateObject("aims.SimpleLabelRenderer")
set mClone = mSimpleLabelRenderer.Clone()
mSimpleLabelRenderer.Field = "CNTRY_NAME"
mSimpleLabelRenderer.FWeight = imsWeight.imsNo_weight
mSimpleLabelRenderer.HMLabels = imsHMLabels.imsOne_label_per_name
mSimpleLabelRenderer.LabelPriorities = "1,1,1,1,2,2,2,2"
mSimpleLabelRenderer.LLPosition = imsLLPosition.imsPlaceNone
mSimpleLabelRenderer.RotationalAngles = 30
mSimpleLabelRenderer.Symbol = mTextSymbol
mMap.Refresh()
```

See Also

imsLLPosition

SimpleLabelRenderer.LWeight property

Description

Constant used to prioritize the importance of labels. The default value is imsNo_weight. The possible values are:

0 = imsNo_weight
1 = imsMed_weight
2 = imsHigh_weight

Write/Read.

Type

Long

Example

```
Dim mSimpleLabelRenderer
Set mSimpleLabelRenderer = Server.CreateObject("aims.SimpleLabelRenderer")
mSimpleLabelRenderer.LWeight = imsWeight.imsNo_weight
```

See Also

imsWeight

SimpleLabelRenderer.RotationalAngles property

Description

Angle at which labels will be placed relative to the feature. Write/Read.

Type

Long

Example

```
dim mClone
dim mSimpleLabelRenderer
set mSimpleLabelRenderer = Server.CreateObject("aims.SimpleLabelRenderer")
set mClone = mSimpleLabelRenderer.Clone()
mSimpleLabelRenderer.Field = "CNTRY_NAME"
mSimpleLabelRenderer.FWeight = imsWeight.imsNo_weight
mSimpleLabelRenderer.HMLabels = imsHMLabels.imsOne_label_per_name
mSimpleLabelRenderer.LabelPriorities = "1,1,1,1,2,2,2,2"
mSimpleLabelRenderer.LLPosition = imsLLPosition.imsPlaceNone
mSimpleLabelRenderer.RotationalAngles = 30
mSimpleLabelRenderer.Symbol = mTextSymbol
mMap.Refresh()
```

SimpleLabelRenderer.Symbol property

Description

Symbol object for a SimpleLabelRenderer. The symbol can be a ShieldSymbol, TextSymbol, CalloutMarkerSymbol, or RasterShieldSymbol. Write/Read.

Type

Object

Example

```
dim mClone
dim mSimpleLabelRenderer
set mSimpleLabelRenderer = Server.CreateObject("aims.SimpleLabelRenderer")
set mClone = mSimpleLabelRenderer.Clone()
mSimpleLabelRenderer.Field = "CNTRY_NAME"
mSimpleLabelRenderer.FWeight = imsWeight.imsNo_weight
mSimpleLabelRenderer.HMLabels = imsHMLabels.imsOne_label_per_name
mSimpleLabelRenderer.LabelPriorities = "1,1,1,1,2,2,2,2"
mSimpleLabelRenderer.LLPosition = imsLLPosition.imsPlaceNone
mSimpleLabelRenderer.RotationalAngles = 30
mSimpleLabelRenderer.Symbol = mTextSymbol
mMap.Refresh()
```

See Also

CalloutMarkerSymbol methods and properties
RasterShieldSymbol methods and properties
ShieldSymbol methods and properties
TextSymbol methods and properties

SimpleLineSymbol.Antialiasing property

Description

Boolean value indicating whether or not antialiasing will be used while rendering the SimpleLineSymbol. Antialiasing allows lines to appear smooth by coloring adjacent pixels in such a way as to give the edge of a line a more gradual fade into the background. The default value is False. Write/Read.

Type

Boolean

Example

```
Dim mSimpleLineSymbol
Set mSimpleLineSymbol = Server.CreateObject("aims.SimpleLineSymbol")
mSimpleLineSymbol.Antialiasing = True
```

SimpleLineSymbol.CapStyle property

Description

CapStyle constant that determines the line end style. The default value is imsRound (0). The possible values are:

0 = imsRound
1 = imsButt
2 = imsSquare

Write/Read

Type

Long

Example

```
Dim mSimpleLineSymbol
Set mSimpleLineSymbol = Server.CreateObject("aims.SimpleLineSymbol")
mSimpleLineSymbol.CapStyle = imsCapStyle.imsRound
```

See Also

imsCapStyle

SimpleLineSymbol.Clone method

Description
Clones the SimpleLineSymbol.

Syntax
SimpleLineSymbol.Clone()

Arguments
None

Returned Value
SimpleLineSymbol Cloned symbol.

Example
```
dim mClone
dim mSimpleLineSymbol
Set mSimpleLineSymbol = Server.CreateObject("aims.SimpleLineSymbol")
Set mClone = mSimpleLineSymbol.Clone()
```

SimpleLineSymbol.Color property

Description
Symbol color constant for the SimpleLineSymbol. The default value is imsBlack (0). Write/Read.

Type
Long

Example
```
Dim mSimpleLineSymbol
Set mSimpleLineSymbol = Server.CreateObject("aims.SimpleLineSymbol")
mSimpleLineSymbol.Color = imsColor.imsRed
```

See Also
imsColor

SimpleLineSymbol.JoinStyle property

Description

Join style constant that determines how intersecting (or joined) lines are displayed with SimpleLineSymbols. The default value is imsRound (0). The possible values are:

0 = imsRound
1 = imsMiter
2 = imsBevel

Write/Read

Type

Long

Example

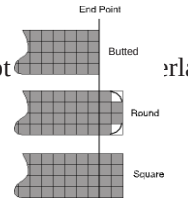
```
Dim mSimpleLineSymbol
Set mSimpleLineSymbol = Server.CreateObject("aims.SimpleLineSymbol")
mSimpleLineSymbol.JoinStyle = imsJoinStyle.imsRound
```

See Also

imsJoinStyle

SimpleLineSymbol.Overlap property

Description



Boolean value indicating whether or not erlap a SimpleLineSymbol. Write/Read.

Type

Boolean

Example

```
Dim mSimpleLineSymbol
Set mSimpleLineSymbol = Server.CreateObject("aims.SimpleLineSymbol")
mSimpleLineSymbol.Overlap = False
```

SimpleLineSymbol.Style property

Description

Constant that determines a line style for a SimpleLineSymbol. The default value is imsSolid (0). The possible values are:

0 = imsSolid
1 = imsDash
2 = imsDot
3 = imsDash_dot
4 = imsDash_dot_dot

Write/Read.

Type

Long

Example

```
Dim mSimpleLineSymbol
Set mSimpleLineSymbol = Server.CreateObject("aims.SimpleLineSymbol")
mSimpleLineSymbol.Style = imsLineStyle.imsSolid
```

See Also

imsLineStyle

SimpleLineSymbol.Transparency property

Description

Transparency coefficient for the SimpleLineSymbol. Lower numbers will display the symbol with greater transparency. The default value is 1.0. Write/Read.

Type

Double

Example

```
Dim mSimpleLineSymbol
Set mSimpleLineSymbol = Server.CreateObject("aims.SimpleLineSymbol")
mSimpleLineSymbol.Transparency = 1.0
```

SimpleLineSymbol.Width property

Description

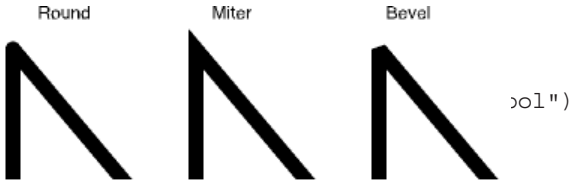
Line width for a SimpleLineSymbol. Write/Read.

Type

Double

Example

```
Dim mSimpleLineSymbol
Set mSimpleLineSymbol = Serv                                    ol")
mSimpleLineSymbol.Width = 2
```



SimpleMarkerSymbol.Antialiasing property

Description

Boolean value indicating whether or not antialiasing will be used while rendering the SimpleMarkerSymbol. Antialiasing allows lines to appear smooth by coloring adjacent pixels in such a way as to give the edge of a line a more gradual fade into the background. The default value is False.

Write/Read

Type

Boolean

Example

```
Dim mSimpleMarkerSymbol
Set mSimpleMarkerSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
mSimpleMarkerSymbol.Antialiasing = True
```

SimpleMarkerSymbol.Clone method

Description

Clones the SimpleMarkerSymbol.

Syntax

SimpleMarkerSymbol.Clone()

Arguments

None

Returned Value

SimpleMarkerSymbol Cloned symbol.

Example

```
dim mClone
dim mSimpleMarkerSymbol
Set mSimpleMarkerSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
Set mClone = mSimpleMarkerSymbol.Clone()
```

SimpleMarkerSymbol.Color property

Description

Symbol color constant for the SimpleMarkerSymbol. The default value is imsBlack (0). Write/Read.

Type

Long

Example

```
Dim mSimpleMarkerSymbol
Set mSimpleMarkerSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
mSimpleMarkerSymbol.Color = imsColor.imsRed
```

See Also

imsColor

SimpleMarkerSymbol.MarkerType property

Description

Marker Symbol type constant for a SimpleMarkerSymbol. The default value is imsCircle (0). The possible values are:

0 = imsCircle
1 = imsTriangle
2 = imsSquare
3 = imsCross
4 = imsStar

Write/Read.

Type

Long

Example

```
Dim mSimpleMarkerSymbol
Set mSimpleMarkerSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
mSimpleMarkerSymbol.MarkerType = imsMarkerType.imsCross
```

See Also

Change_Marker_Symbol.asp
imsMarkerType

SimpleMarkerSymbol.Outline property

Description

Color constant for the outline color of a SimpleMarkerSymbol. The default value is imsBlack (0). Write/Read.

Type

Long

Example

```
Dim mSimpleMarkerSymbol
Set mSimpleMarkerSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
mSimpleMarkerSymbol.Outline = imsColor.imsRed
```

See Also

Change_Marker_Symbol.asp

SimpleMarkerSymbol.Overlap property

Description

Boolean value indicating whether or not labels will overlap a SimpleMarkerSymbol. The default value is True. Write/Read.

Type

Boolean

Example

```
Dim mSimpleMarkerSymbol
Set mSimpleMarkerSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
mSimpleMarkerSymbol.Overlap = False
```

SimpleMarkerSymbol.Shadow property

Description

Color constant for the shadow color of a SimpleMarkerSymbol. The default value is imsBlack (0). Write/Read.

Type

Long

Example

```
Dim mSimpleMarkerSymbol
Set mSimpleMarkerSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
mSimpleMarkerSymbol.Shadow = imsColor.imsBlack
```

SimpleMarkerSymbol.Transparency property

Description

Transparency coefficient for the SimpleMarkerSymbol. Lower numbers will display the symbol with greater transparency. The default value is 1.0. Write/Read.

Type

Double

Example

```
Dim mSimpleMarkerSymbol
Set mSimpleMarkerSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
mSimpleMarkerSymbol.Transparency = 1.0
```

SimpleMarkerSymbol.Width property

Description

Width of a SimpleMarkerSymbol. Write/Read.

Type

Double

Example

```
Dim mSimpleMarkerSymbol
Set mSimpleMarkerSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
mSimpleMarkerSymbol.Width = 2
```

SimplePolygonSymbol.Antialiasing property

Description

Boolean value indicating whether or not antialiasing will be used while rendering the SimplePolygonSymbol. Antialiasing allows lines to appear smooth by coloring adjacent pixels in such a way as to give the edge of a line a more gradual fade into the background. The default value is False. Write/Read.

Type

Boolean

Example

```
Dim mSimplePolygonSymbol
Set mSimplePolygonSymbol = Server.CreateObject("aims.SimplePolygonSymbol")
mSimplePolygonSymbol.Antialiasing = True
```

SimplePolygonSymbol.Boundary property

Description

Boolean value that determines whether or not to draw a boundary symbol for the SimplePolygonSymbol. The default value is True. Write/Read.

Type

Boolean

Example

```
Dim mSimplePolygonSymbol
Set mSimplePolygonSymbol = Server.CreateObject("aims.SimplePolygonSymbol")
mSimplePolygonSymbol.Boundary = False
```

See Also

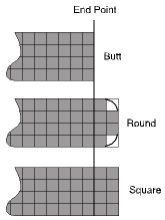
Change_Polygon_Symbol.asp

SimplePolygonSymbol.BoundaryCapType property

Description

Returns or sets a boundary line ending constant that determines how the boundary ends will be displayed. The default value is imsButt (1). The possible values are:

0 = imsRound
1 = imsButt
2 = imsSquare



Write/Read

Type

Long

Example

```
Dim mSimplePolygonSymbol
Set mSimplePolygonSymbol = Server.CreateObject("aims.SimplePolygonSymbol")
mSimplePolygonSymbol.BoundaryCapType = imsCapStyle.imsRound
```

See Also

imsCapStyle

SimplePolygonSymbol.BoundaryColor property

Description

Color constant for the boundary color of a SimplePolygonSymbol. The default value is imsBlack (0). Write/Read

Type

Long

Example

```
Dim mSimplePolygonSymbol
Set mSimplePolygonSymbol = Server.CreateObject("aims.SimplePolygonSymbol")
mSimplePolygonSymbol.BoundaryColor = imsColor.imsBlack
```

See Also

Change_Polygon_Symbol.asp

SimplePolygonSymbol.BoundaryJoinType property

Description

Join style constant that determines how intersecting (or joined) boundary lines are displayed with SimplePolygonSymbols. The default value is imsRound (0). The possible values are:

0 = imsRound
1 = imsMiter
2 = imsBevel

Write/Read

Type

Long

Example

```
Dim mSimplePolygonSymbol
Set mSimplePolygonSymbol = Server.CreateObject("aims.SimplePolygonSymbol")
mSimplePolygonSymbol.JoinStyle = imsJoinStyle.imsRound
```

See Also

imsJoinStyle

SimplePolygonSymbol.BoundaryStyle property

Description

Constant that determines the boundary style for a SimplePolygonSymbol boundary. The default value is imsSolid (0). The possible values are:

0 = imsSolid
1 = imsDash
2 = imsDot
3 = imsDash_dot
4 = imsDash_dot_dot

Write/Read

Type

Long

Example

```
Dim mSimplePolygonSymbol
Set mSimplePolygonSymbol = Server.CreateObject("aims.SimplePolygonSymbol")
mSimplePolygonSymbol.BoundaryStyle = imsLineStyle.imsSolid
```

See Also

Change_Polygon_Symbol.asp
imsLineStyle

SimplePolygonSymbol.BoundaryTransparency property

Description

Boundary transparency coefficient for the SimplePolygonSymbol. Lower numbers will display the symbol with greater transparency. The default value is 1.0. Write/Read

Type

Double

Example

```
Dim mSimplePolygonSymbol
Set mSimplePolygonSymbol = Server.CreateObject("aims.SimplePolygonSymbol")
mSimplePolygonSymbol.BoundaryTransparency = 1.0
```

SimplePolygonSymbol.BoundaryWidth property

Description

Boundary width of the SimplePolygonSymbol. The default value is 0. Write/Read

Type

Double

Example

```
Dim mSimplePolygonSymbol
Set mSimplePolygonSymbol = Server.CreateObject("aims.SimplePolygonSymbol")
mSimplePolygonSymbol.BoundaryWidth = 1
```

See Also

Change_Polygon_Symbol.asp

SimplePolygonSymbol.Clone method

Description
Clones the SimplePolygonSymbol.

Syntax
SimplePolygonSymbol.Clone()

Arguments
None

Returned Value
SimplePolygonSymbol Cloned symbol.

Example
```
dim mClone
dim mSimplePolygonSymbol
Set mSimplePolygonSymbol = Server.CreateObject("aims.SimplePolygonSymbol")
Set mClone = mSimplePolygonSymbol.Clone()
```

SimplePolygonSymbol.FillColor property

Description
Color constant for the fill color of a SimplePolygonSymbol. The default value is imsBlack (0). Write/Read.

Type
Long

Example
```
Dim mSimplePolygonSymbol
Set mSimplePolygonSymbol = Server.CreateObject("aims.SimplePolygonSymbol")
mSimplePolygonSymbol.FillColor = imsColor.imsRed
```

See Also
Change_Polygon_Symbol.asp

SimplePolygonSymbol.FillInterval property

Description
Fill interval for hatch fills. The default value is 6. Write/Read.

Type
Double

Example
```
Dim mSimplePolygonSymbol
Set mSimplePolygonSymbol = Server.CreateObject("aims.SimplePolygonSymbol")
mSimplePolygonSymbol.FillInterval = 10
```

See Also
Change_Polygon_Symbol.asp

SimplePolygonSymbol.FillStyle property

Description
Constant indicating the fill style for the SimplePolygonSymbol. The default value is imsSolid (0). The possible values are:

0 = imsSolid
1 = imsBDiagonal
2 = imsFDiagonal
3 = imsCross
4 = imsDiagcross

Write/Read.

Type
Long

Example
```
Dim mSimplePolygonSymbol
Set mSimplePolygonSymbol = Server.CreateObject("aims.SimplePolygonSymbol")
mSimplePolygonSymbol.FillStyle = imsFillStyle.imsSolid
```

See Also
Change_Polygon_Symbol.asp
imsFillStyle

SimplePolygonSymbol.FillTransparency property

Description
Value to set percentage of transparency for the polygon fill. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent. Default is 1.0.

Type
Double

SimpleRenderer.Clone method

Description
Clones the SimpleRenderer.

Syntax
SimpleRenderer.Clone()

Arguments
None

Returned Value
SimpleRenderer Cloned renderer.

Example
```
Dim mClone
Dim mSimpleRenderer
Set mSimpleRenderer = Server.CreateObject("aims.SimpleRenderer")
Set mClone = mSimpleRenderer.Clone()
```

SimpleRenderer.Symbol property

Description
Symbol object for a renderer. It can be a TrueTypeMarkerSymbol, SimpleMarkerSymbol, RasterMarkerSymbol, HashLineSymbol, SimpleLineSymbol, RasterFillSymbol, SimplePolygonSymbol, or GradientFillSymbol. Write/Read.

Type
Object

Example
```
Dim mSimpleRenderer
Dim mSimpleMarkerSymbol
Set mSimpleRenderer = Server.CreateObject("aims.SimpleRenderer")
Set mSimpleMarkerSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
mSimpleRenderer.Symbol = mSimpleMarkerSymbol
```

See Also
Buffer.asp
Change_Marker_Symbol.asp
Change_Polygon_Symbol.asp
GradientFillSymbol methods and properties
HashLineSymbol methods and properties
RasterFillSymbol methods and properties
RasterMarkerSymbol methods and properties
SimpleLineSymbol methods and properties
SimpleMarkerSymbol methods and properties
SimplePolygonSymbol methods and properties
TrueTypeMarkerSymbol methods and properties

TableDesc.Count property

Description
Number of described fields. Read only.

Type
Long

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mTableDesc
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSServer.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"
Set mTableDesc = mMap.Layers.Item(2).Recordset.TableDesc
mCount = mTableDesc.Count
Response.write mCount
```

5 = imsHorizontal
6 = imsVertical
7 = imsLightGray
8 = imsGray

See Also
Query_Attribute_Data.asp

TableDesc.FieldLength method

Description
Returns the length of the field by index.

Syntax
TableDesc.FieldLength(index as Long)

Arguments

Index	Long	Index to reference to the field.

Returned Value

Long	Length of the field.

Example
```
Dim mArcIMSConnector
```

```
Dim mMap
Dim mTableDesc
Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.Init mArcIMSConnector, "IMSMapService"
Set mTableDesc = mMap.Layers.Item(2).Recordset.TableDesc

for I = 1 to mTableDesc.Count
   mFieldLength = mTableDesc.FieldLength(i)
next
```

See Also

TableDesc.FieldName
TableDesc.FieldPrecision
TableDesc.FieldType

TableDesc.FieldName method

Description

Returns the name of the field by index.

Syntax

TableDesc.FieldName(index as Long)

Arguments

Index	Long	Index to reference to the field.

Returned Value

String	Name of the field.

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mTableDesc

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mTableDesc = mMap.Layers.Item(2).Recordset.TableDesc
For I = 1 to mTableDesc.Count
   mTableDesc.FieldName(i)
Next
```

See Also

Query_Attribute_Data.asp

TableDesc.FieldPrecision method

Description

Returns the precision of the field by index.

Syntax

TableDesc.FieldPrecision(index as Long)

Arguments

Index	Long	Index to reference to the field.

Returned Value

Long	Precision of the field.

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mTableDesc

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mTableDesc = mMap.Layers.Item(2).Recordset.TableDesc

For I = 1 to mTableDesc.Count
   mFieldPrecision = mTableDesc.FieldPrecision(i)
Next
```

See Also

TableDesc.FieldLength
TableDesc.FieldName
TableDesc.FieldType

TableDesc.FieldType method

Description
Returns the type of the field by index.

Syntax
TableDesc.FieldType(index as Long)

Arguments
Index	Long	Index to reference to the field.

Returned Value
Long	Type of field. Use imsFieldType constants.

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mTableDesc

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300

Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set mTableDesc = mMap.Layers.Item(2).Recordset.TableDesc

For I = 1 to mTableDesc.Count
   mTableDesc.FieldType(i)
Next
```

See Also
TableDesc.FieldLength
TableDesc.FieldName
TableDesc.FieldPrecision

TextMarkerSymbol.Angle property

Description

Angle of rotation for a TextMarkerSymbol in degrees going counterclockwise; 0 degrees is horizontal. The default value is 0. Write/Read.

Type

Double

Example

```
Dim mTextMarkerSymbol
Set mTextMarkerSymbol = Server.CreateObject("aims.TextMarkerSymbol")
mTextMarkerSymbol.Angle = 30
```

TextMarkerSymbol.Antialiasing property

Description

Boolean value indicating whether or not antialiasing will be used while rendering the TextMarkerSymbol. Antialiasing allows lines to appear smooth by coloring adjacent pixels in such a way as to give the edge of a line a more gradual fade into the background. The default value is False. Write/Read.

Type

Boolean

Example

```
Dim mTextMarkerSymbol
Set mTextMarkerSymbol = Server.CreateObject("aims.TextMarkerSymbol")
mTextMarkerSymbol.Antialiasing = True
```

TextMarkerSymbol.Blockout property

Description
Constant for the background color of a TextMarkerSymbol. The default value is imsBlack (0). Write/Read

Type
Long

Example
```
Dim mTextMarkerSymbol
Set mTextMarkerSymbol = Server.CreateObject("aims.TextMarkerSymbol")
mTextMarkerSymbol.Blockout = imsColor.imsRed
```

See Also
imsColor

TextMarkerSymbol.Clone method

Description
Clones the TextMarkerSymbol.

Syntax
TextMarkerSymbol.Clone()

Arguments
None

Returned Value
TextMarkerSymbol Cloned symbol.

Example
```
dim mClone
dim mTextMarkerSymbol
Set mTextMarkerSymbol = Server.CreateObject("aims.TextMarkerSymbol")
Set mClone = mTextMarkerSymbol.Clone()
```

TextMarkerSymbol.Font property

Description

Name of the font used with the TextMarkerSymbol. The default value is Arial. Write/Read.

Type

String

Example

```
Dim mTextMarkerSymbol
Set mTextMarkerSymbol = Server.CreateObject("aims.TextMarkerSymbol")
mTextMarkerSymbol.Font = "Arial"
```

TextMarkerSymbol.FontColor property

Description

Font color constant for the font used with a TextMarkerSymbol. The default value is imsBlack (0). Write/Read.

Type

Long

Example

```
Dim mTextMarkerSymbol
Set mTextMarkerSymbol = Server.CreateObject("aims.TextMarkerSymbol")
mTextMarkerSymbol.Color = imsColor.imsRed
```

TextMarkerSymbol.FontSize property

Description

Font used with a TextMarkerSymbol. The default value is 12. Write/Read.

Type

Long

Example

```
Dim mTextMarkerSymbol
Set mTextMarkerSymbol = Server.CreateObject("aims.TextMarkerSymbol")
mTextMarkerSymbol.FontSize = 10
```

TextMarkerSymbol.FontStyle property

Description

Font style constant for the font used with a TextMarkerSymbol. The default value is imsRegular (0). The possible values are:

0 = imsRegular
1 = imsBold
2 = imsItalic
3 = imsUnderline
4 = imsOutline

Write/Read.

Type

Long

Example

```
Dim mTextMarkerSymbol
Set mTextMarkerSymbol = Server.CreateObject("aims.TextMarkerSymbol")
mTextMarkerSymbol.FontStyle = imsFontStyle.imsBold
```

TextMarkerSymbol.Glowing property

Description

Glowing color constant for a TextMarkerSymbol. Glowing is the halo effect around text labels. The default value is imsBlack (0). Write/Read.

Type

Long

Example

```
Dim mTextMarkerSymbol
Set mTextMarkerSymbol = Server.CreateObject("aims.TextMarkerSymbol")
mTextMarkerSymbol.Glowing = imsColor.imsRed
```

TextMarkerSymbol.HAlignment property

Description

Constant indicating the horizontal alignment of the label compared to the label point. The default value is imsRight (2). The possible values are:

0 = imsLeft
1 = imsCenter
2 = imsRight

Write/Read.

Type

Long

Example

```
Dim mTextMarkerSymbol
Set mTextMarkerSymbol = Server.CreateObject("aims.TextMarkerSymbol")
mTextMarkerSymbol.HAlignment = imsHAlignment.imsLeft
```

See Also

imsHAlignment

TextMarkerSymbol.Interval property

Description

Distance in pixels between a feature and the printed label. The default value is 0. Write/Read.

Type

Double

Example

```
Dim mTextMarkerSymbol
Set mTextMarkerSymbol = Server.CreateObject("aims.TextMarkerSymbol")
mTextMarkerSymbol.Interval = 10
```

TextMarkerSymbol.Outline property

Description

Color constant for the outline color of a TextMarkerSymbol. The default value is imsBlack (0). Write/Read.

Type

Long

Example

```
Dim mTextMarkerSymbol
Set mTextMarkerSymbol = Server.CreateObject("aims.TextMarkerSymbol")
mTextMarkerSymbol.Outline = imsColor.imsRed
```

TextMarkerSymbol.PrintMode property

Description

PrintMode constant that determines how the text for a TextMarkerSymbol will be rendered. The default value is imsNone (0). The possible values are:

0 = imsNone
1 = imsTitleCaps
2 = imsAllUpper
3 = imsAllLower

Write/Read.

Type

Long

Example

```
Dim mTextMarkerSymbol
Set mTextMarkerSymbol = Server.CreateObject("aims.TextMarkerSymbol")
mTextMarkerSymbol.PrintMode = imsPrintMode.imsNone
```

See Also

imsPrintMode

TextMarkerSymbol.Shadow property

Description

Color constant for the shadow color of a TextMarkerSymbol. The default value is imsBlack (0). Write/Read.

Type

Long

Example

```
Dim mTextMarkerSymbol
Set mTextMarkerSymbol = Server.CreateObject("aims.TextMarkerSymbol")
mTextMarkerSymbol.Shadow = imsColor.imsBlack
```

TextMarkerSymbol.Transparency property

Description

Transparency coefficient for the TextMarkerSymbol. Lower numbers will display the symbol with greater transparency. The default value is 1.0. Write/Read.

Type

Double

Example

```
Dim mTextMarkerSymbol
Set mTextMarkerSymbol = Server.CreateObject("aims.TextMarkerSymbol")
mTextMarkerSymbol.Transparency = 1.0
```

TextMarkerSymbol.VAlignment property

Description

Constant indicating the vertical alignment of a TextMarkerSymbol compared to a label point. The default value is imsTop (0). The possible values are:

0 = imsTop
1 = imsCenter
2 = imsBottom

Write/Read

Type

Long

Example

```
Dim mTextMarkerSymbol
Set mTextMarkerSymbol = Server.CreateObject("aims.TextMarkerSymbol")
mTextMarkerSymbol.VAlignment= imsVAlignment.imsTop
```

See Also

imsVAlignment

TextObject.Id property

Description

Contains the ID for the Text object. Write/Read.

Type

String

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mTableDesc
Dim al
Dim text
Dim mId

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set al = Server.CreateObject("aims.AcetateLayer")
al.Visible = true

Set text = Server.CreateObject("aims.TextObject")
mId = text.Id
Response.write mId
```

TextObject.Label property

Description

Contains the label for the Text object. Write/Read.

Type

String

Example

```
Dim mArcIMSConnector
Dim mMap
Dim mTableDesc
Dim al
Dim text
Dim mId

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set al = Server.CreateObject("aims.AcetateLayer")
al.Visible = true

Set text = Server.CreateObject("aims.TextObject")
text.Label = "ActiveXConnector"
```

TextObject.Name property

Description
TextObject's name property. Write/Read.

Type
String

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mTableDesc
Dim al
Dim text
Dim mId

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set al = Server.CreateObject("aims.AcetateLayer")
al.Visible = true

Set text = Server.CreateObject("aims.TextObject")
text.Name = "ActiveXConnector"
```

TextObject.TextMarkerSymbol property

Description
Specifies how to depict text. Write/Read.

Type
TextMarkerSymbol

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mTableDesc
Dim al
Dim text
Dim mId

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set al = Server.CreateObject("aims.AcetateLayer")
al.Visible = true

Set text = Server.CreateObject("aims.TextObject")
text.TextMarkerSymbol.Color = 255
```

TextObject.X property

Description
Contains the X coordinate of the text. Write/Read

Type
Double

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mTableDesc
Dim al
Dim text
Dim mId

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set al = Server.CreateObject("aims.AcetateLayer")
al.Visible = true

Set text = Server.CreateObject("aims.TextObject")
text.X = 56
```

TextObject.Y property

Description
Contains the Y coordinate of the text. Write/Read

Type
Double

Example
```
Dim mArcIMSConnector
Dim mMap
Dim mTableDesc
Dim al
Dim text
Dim mId

Set mArcIMSConnector = Server.CreateObject("aims.ArcIMSConnector")
mArcIMSConnector.ServerName = "IMSServer"
mArcIMSConnector.ServerPort = 5300
Set mMap = Server.CreateObject("aims.Map")
mMap.InitMap mArcIMSConnector, "IMSMapService"

Set al = Server.CreateObject("aims.AcetateLayer")
al.Visible = true

Set text = Server.CreateObject("aims.TextObject")
text.Y = 46
```

TextSymbol.Antialiasing property

Description
Boolean value indicating whether or not antialiasing will be used while rendering the TextSymbol. Antialiasing allows lines to appear smooth by coloring adjacent pixels in such a way as to give the edge of a line a more gradual fade into the background. The default value is False. Write/Read.

Type
Boolean

Example
```
Dim mTextSymbol
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mTextSymbol.Antialiasing = True
```

TextSymbol.Blockout property

Description
Constant for the background color of a TextSymbol. The default value is imsBlack (0). Write/Read.

Type
Long

Example
```
Dim mTextSymbol
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mTextSymbol.Blockout = imsColor.imsRed
```

See Also
imsColor

TextSymbol.Clone method

Description
Clones the TextSymbol.

Syntax
TextSymbol.Clone()

Arguments
None

Returned Value
TextSymbol Cloned symbol.

Example
```
dim mClone
dim mTextSymbol
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
Set mClone = mTextSymbol.Clone()
```

TextSymbol.Font property

Description
Name of the font used with the TextSymbol. Font name. The default value is Arial. Write/Read.

Type
String

Example
```
Dim mTextSymbol
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mTextSymbol.Font = "Arial"
```

TextSymbol.FontColor property

Description
Font color constant for the font used with a TextSymbol. The default value is imsBlack (0). Write/Read.

Type
Long

Example
```
Dim mTextSymbol
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mTextSymbol.Color = imsColor.imsRed
```

TextSymbol.FontSize property

Description
Font size for the font used with a TextSymbol. The default value is 12. Write/Read.

Type
Long

Example
```
Dim mTextSymbol
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mTextSymbol.FontSize = 10
```

TextSymbol.FontStyle property

Description

Font style constant for the font used with a TextSymbol. The default value is imsRegular (0). The possible values are:

0 = imsRegular
1 = imsBold
2 = imsItalic
3 = imsUnderline
4 = imsOutline

Write/Read

Type

Long

Example

```
Dim mTextSymbol
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mTextSymbol.FontStyle = imsFontStyle.imsBold
```

TextSymbol.Glowing property

Description

Glowing color constant for a TextSymbol. Glowing is the halo effect around text labels. The default value is imsBlack (0). Write/Read

Type

Long

Example

```
Dim mTextSymbol
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mTextSymbol.Glowing = imsColor.imsRed
```

TextSymbol.Interval property

Description

Distance in pixels between a feature and the printed label. The default value is 0. Write/Read

Type

Double

Example

```
Dim mTextSymbol
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mTextSymbol.Interval = 10
```

TextSymbol.Outline property

Description

Color constant for the outline color of a TextSymbol. The default value is imsBlack (0). Write/Read.

Type

Long

Example

```
Dim mTextSymbol
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mTextSymbol.Outline = imsColor.imsRed
```

TextSymbol.PrintMode property

Description

PrintMode constant that determines how the text for a TextSymbol will be rendered. The default value is imsNone (0). The possible values are:

0 = imsNone
1 = imsTitleCaps
2 = imsAllUpper
3 = imsAllLower

Write/Read.

Type

Long

Example

```
Dim mTextSymbol
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mTextSymbol.PrintMode = imsPrintMode.imsNone
```

See Also

imsPrintMode

TextSymbol.Shadow property

Description
Color constant for the shadow color of a TextSymbol. The default value is imsBlack (0). Write/Read.

Type
Long

Example
```
Dim mTextSymbol
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mTextSymbol.Shadow = imsColor.imsBlack
```

TextSymbol.Transparency property

Description
Transparency coefficient for the TextSymbol. Lower numbers will display the symbol with greater transparency. The default value is 1.0. Write/Read.

Type
Double

Example
```
Dim mTextSymbol
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mTextSymbol.Transparency = 1.0
```

TrueTypeMarkerSymbol.Angle property

Description

Angle of rotation for a TrueTypeMarkerSymbol in degrees going counterclockwise; 0 degrees is horizontal. The default value is 0. Write/Read

Type

Double

Example

```
Dim mTrueTypeMarkerSymbol
Set mTrueTypeMarkerSymbol = Server.CreateObject("aims.TrueTypeMarkerSymbol")
mTrueTypeMarkerSymbol.Angle = 30
```

TrueTypeMarkerSymbol.Antialiasing property

Description

Boolean value indicating whether or not antialiasing will be used while rendering the TrueTypeMarkerSymbol. Antialiasing allows lines to appear smooth by coloring adjacent pixels in such a way as to give the edge of a line a more gradual fade into the background. The default value is False. Write/Read

Type

Boolean

Example

```
Dim mTrueTypeMarkerSymbol
Set mTrueTypeMarkerSymbol = Server.CreateObject("aims.TrueTypeMarkerSymbol")
mTrueTypeMarkerSymbol.Antialiasing = True
```

TrueTypeMarkerSymbol.Character property

Description

Text character index in a font containing the symbol to use in a decimal notation. Available values are 32–255. Write/Read

Type

Long

Example

```
Dim mTrueTypeMarkerSymbol
Set mTrueTypeMarkerSymbol = Server.CreateObject("aims.TrueTypeMarkerSymbol")
mTrueTypeMarkerSymbol.Character = 35
```

TrueTypeMarkerSymbol.Clone method

Description
Clones the TrueTypeMarkerSymbol.

Syntax
TrueTypeMarkerSymbol.Clone()

Arguments
None

Returned Value
TrueTypeMarkerSymbol Cloned symbol.

Example
```
dim mClone
dim mTrueTypeMarkerSymbol
Set mTrueTypeMarkerSymbol = Server.CreateObject("aims.TrueTypeMarkerSymbol")
Set mClone = mTrueTypeMarkerSymbol.Clone()
```

TrueTypeMarkerSymbol.Font property

Description
Name of the font used with the TrueTypeMarkerSymbol. The default value is default. Write/Read

Type
String

Example
```
Dim mTrueTypeMarkerSymbol
Set mTrueTypeMarkerSymbol = Server.CreateObject("aims.TrueTypeMarkerSymbol")
mTrueTypeMarkerSymbol.Font = "Arial"
```

TrueTypeMarkerSymbol.FontColor property

Description
Font color constant for the font used with a TrueTypeMarkerSymbol. The default value is imsBlack (0). Write/Read

Type
Long

Example
```
Dim mTrueTypeMarkerSymbol
Set mTrueTypeMarkerSymbol = Server.CreateObject("aims.TrueTypeMarkerSymbol")
mTrueTypeMarkerSymbol.Color = imsColor.imsRed
```

TrueTypeMarkerSymbol.FontSize property

Description

Font size for the font used with a TrueTypeMarkerSymbol. The default value is 12. Write/Read

Type

Long

Example

```
Dim mTrueTypeMarkerSymbol
Set mTrueTypeMarkerSymbol = Server.CreateObject("aims.TrueTypeMarkerSymbol")
mTrueTypeMarkerSymbol.FontSize = 10
```

TrueTypeMarkerSymbol.FontStyle property

Description

Font style constant for the font used with a TrueTypeMarkerSymbol. The default value is imsRegular (0). The possible values are:

0 = imsRegular
1 = imsBold
2 = imsItalic
3 = imsUnderline
4 = imsOutline

Write/Read

Type

Long

Example

```
Dim mTrueTypeMarkerSymbol
Set mTrueTypeMarkerSymbol = Server.CreateObject("aims.TrueTypeMarkerSymbol")
mTrueTypeMarkerSymbol.FontStyle = imsFontStyle.imsBold
```

TrueTypeMarkerSymbol.Glowing property

Description

Glowing color constant for a TrueTypeMarkerSymbol. Glowing is the halo effect around the symbol. The default value is imsBlack (0). Write/Read

Type

Long

Example

```
Dim mTrueTypeMarkerSymbol
Set mTrueTypeMarkerSymbol = Server.CreateObject("aims.TrueTypeMarkerSymbol")
mTrueTypeMarkerSymbol.Glowing = imsColor.imsRed
```

TrueTypeMarkerSymbol.Outline property

Description

Color constant for the outline color of a TrueTypeMarkerSymbol. The default value is imsBlack (0). Write/Read

Type

Long

Example

```
Dim mTrueTypeMarkerSymbol
Set mTrueTypeMarkerSymbol = Server.CreateObject("aims.TrueTypeMarkerSymbol")
mTrueTypeMarkerSymbol.Outline = imsColor.imsRed
```

TrueTypeMarkerSymbol.Overlap property

Description

Boolean value indicating whether or not labels will overlap a TrueTypeMarkerSymbol. The default value is True. Write/Read

Type

Boolean

Example

```
Dim mTrueTypeMarkerSymbol
Set mTrueTypeMarkerSymbol = Server.CreateObject("aims.TrueTypeMarkerSymbol")
mTrueTypeMarkerSymbol.Overlap = False
```

TrueTypeMarkerSymbol.Shadow property

Description

Color constant for the shadow color of a TrueTypeMarkerSymbol. The default value is imsBlack (0). Write/Read

Type

Long

Example

```
Dim mTrueTypeMarkerSymbol
Set mTrueTypeMarkerSymbol = Server.CreateObject("aims.TrueTypeMarkerSymbol")
mTrueTypeMarkerSymbol.Shadow = imsColor.imsBlack
```

TrueTypeMarkerSymbol.Transparency property

Description

Transparency coefficient for the TrueTypeMarkerSymbol. Lower numbers will display the symbol with greater transparency. The default value is 1.0. Write/Read.

Type

Double

Example

```
Dim mTrueTypeMarkerSymbol
Set mTrueTypeMarkerSymbol = Server.CreateObject("aims.TrueTypeMarkerSymbol")
mTrueTypeMarkerSymbol.Transparency = 1.0
```

ValueMapLabelRenderer.Add method

Description

Adds a symbol representing a value break to the ValueMapLabelRenderer.

Syntax

ValueMapLabelRenderer.Add (Symbol as Object, Range as Long, Value, Lower, Upper, Optional Equality as Long [default = 2], Optional Method as Long [default = 1])

Arguments

Symbol	Object	Symbol object to add.
Range	Variant	Type constant of the range (Exact, Range, or Other).
Value	Variant	Variant value for exact symbol.
Lower	Variant	Lower value for range symbol.
Upper	Variant	Upper value for range symbol.
Equality	Long	Optional value for Equality. Used with imsRange.imsRange.
Method	Long	Optional value for Method. Used with imsRange.imsExact.

Returned Value

Boolean

Example

```
Dim mValueMapLabelRenderer
Dim mTextSymbol
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mTextSymbol.FontColor = imsColor.imsRed
mResult = mValueMapLabelRenderer.Add(mTextSymbol, imsRange.imsExact, 25,0,0,,0)
mTextSymbol.FontColor = imsColor.imsBlue
mResult = mValueMapLabelRenderer.Add(mTextSymbol, imsRange.imsRange, 0,26,31,0,0)
mTextSymbol.FontColor = imsColor.imsGreen
mResult = mValueMapLabelRenderer.Add(mTextSymbol, imsRange.imsOther, 0, 0, 0)
```

See Also

CalloutMarkerSymbol methods and properties
RasterShieldSymbol methods and properties
ShieldSymbol methods and properties
TextSymbol methods and properties

ValueMapLabelRenderer.Clear method

Description
Clears all the class break symbols and values from the ValueMapLabelRenderer.

Syntax
ValueMapLabelRenderer.Clear()

Arguments
None

Returned Value
None

Example
```
Dim mValueMapLabelRenderer
Dim mTextSymbol
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mResult = mValueMapLabelRenderer.Add(mTextSymbol, imsRange.imsExact, 233, 0, 0)
mValueMapLabelRenderer.Clear()
```

ValueMapLabelRenderer.Clone method

Description
Clones the ValueMapLabelRenderer.

Syntax
ValueMapLabelRenderer.Clone()

Arguments
None

Returned Value
ValueMapLabelRenderer Cloned ValueMapLabelRenderer.

Example
```
Dim mClone
Dim mValueMapLabelRenderer
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
Set mClone = mValueMapLabelRenderer.Clone()
```

ValueMapLabelRenderer.Count property

Description
Number of symbols in the ValueMapLabelRenderer. Read only.

Type
Long

Example
```
Dim mValueMapLabelRenderer
Dim mTextSymbol
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mResult = mValueMapLabelRenderer.Add(mTextSymbol, imsRange.imsExact, 233, 0, 0)
mCount = mValueMapLabelRenderer.Count
```

ValueMapLabelRenderer.Equality property

Description
Used in conjunction with imsRange. Determines whether or not the bounds of the range (the bounding values) are included in a set or returned range, as follows:

all—all values in the range are returned or set

lower—all values excluding the highest value in the range are returned or set

upper—all values excluding the lowest value in the range are returned or set

Write/read.

Syntax
Equality(Index as Long) as imsEquality

Argument
The number you provide for the index specifies the position of the equality in the collection.

Type
Long

Example
```
Dim mValueMapLabelRenderer
Dim mTextSymbol
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
mValueMapLabelRenderer.LabelBufferRatio=1.0
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mResult = mValueMapLabelRenderer.Add(mTextSymbol, imsRange.imsRange, 233, 0, 0)
mValueMapLabelRenderer.Equality(1)=imsEquality.imsAll
```

See Also
imsRange

ValueMapLabelRenderer.FWeight property

Description

Prioritizes the importance of features. The feature weight determines how important the feature labeled is for the label placement algorithm.

The possible values are:

0 = imsNo_weight

1 = imsMed_weight

2 = imsHigh_weight

If imsNo_weight is specified, then the feature has no importance and can be labeled over. If imsHigh_weight is specified, then the feature has high importance and cannot be labeled over. Write/Read

Type

Long

Example

```
Dim mValueMapLabelRenderer
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
mValueMapLabelRenderer.FWeight = imsWeight.imsNo_weight
```

See Also

imsWeight

ValueMapLabelRenderer.HMLabels property

Description

Constant value indicating how many labels are to be drawn while labeling features. The default value is imsOne_label_per_name (0). The possible values are:

0 = imsOne_label_per_name

1 = imsOne_label_per_shape

2 = imsOne_label_per_part

Write/Read

Type

Long

Example

```
Dim mSValueMapLabelRenderer
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
mValueMapLabelRenderer.HMLabels = imsHMLabels.imsOne_label_per_name
```

See Also

imsHMLabels

ValueMapLabelRenderer.LabelBufferRatio property

Description

Value used to set a buffer around the label. When this is set, no labels overlap within the buffer range. The ratio is the fraction of the height or the width of the label rectangle (whichever is smaller) compared to the width of the buffer. A ratio of 0.0 means no buffer. A ratio of 1.0 means that the buffer is twice the size of the label (the label width equals the buffer width). A negative ratio causes the buffer to be smaller than the label. This can be used to allow labels to overlap. Write/Read

Type

Double

Example

```
Dim mValueMapLabelRenderer
Dim mTextSymbol
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
mValueMapLabelRenderer.LabelBufferRatio=1.0
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mResult = mValueMapLabelRenderer.Add(mTextSymbol, imsRange.imsExact, 233, 0, 0)
mSymbol = mValueMapLabelRenderer.Symbol(1)
```

ValueMapLabelRenderer.LabelField property

Description

Name of the field containing text for labeling features. Write/Read

Type

String

Example

```
Dim mValueMapLabelRenderer
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
mValueMapLabelRenderer.LabelField = "CNTRY_NAME"
```

ValueMapLabelRenderer.LabelPriorities property

Description

String value that determines where to place the label around the point. There are eight positions for labels to be placed around a point. The LabelPriorities string contains eight comma-delimited priorities for each position. A 0 at any position tells the placement algorithm not to place a label there at all. A value of 1 tells the algorithm to try and place a label at that position first. Write/Read

Type

String

Example

```
Dim mValueMapLabelRenderer
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
mValueMapLabelRenderer.LabelPriorities = "2,2,1,4,5,3,2,4"
```

ValueMapLabelRenderer.LLPosition property

Description

Constant value that determines where to place a label on a line. The default is imsPlaceAbove. Write/Read

Type

Long

Example

```
Dim mValueMapLabelRenderer
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
mValueMapLabelRenderer.LLPosition = imsLLPosition.imsPlaceNone
```

See Also

imsLLPosition

ValueMapLabelRenderer.LookupField property

Description

Name of the field used for specifying ranges for Range or exact values for Exact. Write/Read

Type

String

Example

```
Dim mValueMapLabelRenderer
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
mValueMapLabelRenderer.LookupField = "CNTRY_NAME"
```

ValueMapLabelRenderer.Lower property

Description
Lower value of a range for a symbol in a ValueMapLabelRenderer. Write/Read

Syntax
Lower(Index as Long) as Variant

Argument
The number you provide for the index specifies the position of the lower value in the collection.

Type
Variant

Example
```
Dim mValueMapLabelRenderer
Dim mTextSymbol
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mResult = mValueMapLabelRenderer.Add(mTextSymbol, imsRange.imsRange, 0, 10, 31)
mLower = mValueMapLabelRenderer.Lower(1)
```

ValueMapLabelRenderer.LWeight property

Description
Value indicating the importance of labels for a feature. If the labels are more important than the features themselves, keep LWeight set to imsHigh_weight. The default value is imsNo_weight. Possible values are:

0 = imsNo_weight
1 = imsMed_weight
2 = imsHigh_weight

Write/Read

Type
Long

Example
```
Dim mValueMapLabelRenderer
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
mValueMapLabelRenderer.LWeight = imsWeight.imsNo_weight
```

See Also
imsWeight

ValueMapLabelRenderer.Method property

Description

Used in conjunction with imsRange.imsExact. Returns or sets the way a value in the data field is compared to the Exact value. Use imsMethod.imsExact for an exact match. Use imsMethod.isContained to search for the value anywhere in a string. String comparisons are case sensitive. Write/Read

Syntax

Method(Index as Long) as imsMethod

Argument

The number you provide for the index specifies the position of the method in the collection.

Type

Long

Example

```
Dim mValueMapLabelRenderer
Dim mTextSymbol
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
mValueMapLabelRenderer.LabelBufferRatio = 1.0
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mResult = mValueMapLabelRenderer.Add(mTextSymbol, imsRange.imsExact, 233, 0, 0)
mValueMapLabelRenderer.Method(1) = imsMethod.imsIsExact
```

See Also

imsMethod
imsRange

ValueMapLabelRenderer.Range property

Description

Type of the range (Exact, Range, or Other) for a specified symbol in a ValueMapLabelRenderer. The possible values are:

0 = imsExact
1 = imsRange
2 = imsOther

Write/Read

Syntax

Range(Index as Long) as imsRange

Argument

The number you provide for the index specifies the position of the range in the collection.

Type

Long

Example

```
Dim mValueMapLabelRenderer
Dim mTextSymbol
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mResult = mValueMapLabelRenderer.Add(mTextSymbol, imsRange.imsExact, 233, 0, 0)
mRange = mValueMapLabelRenderer.Range(1)
```

See Also

imsRange

ValueMapLabelRenderer.Remove method

Description

Remove a symbol from the ValueMapLabelRenderer symbol collection by the index. Write/Read.

Syntax

ValueMapLabelRenderer.Remove (index as Long)

Argument

Index Long The index of a symbol in the symbol collection of the ValueMapLabelRenderer.

Returned Value

Boolean

Example

```
Dim mValueMapLabelRenderer
Dim mTextSymbol
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mResult = mValueMapLabelRenderer.Add(mTextSymbol, imsRange.imsExact, 233, 0, 0)
mResult = mValueMapLabelRenderer.Remove(1)
```

ValueMapLabelRenderer.RotationalAngles property

Description
Possible angles at which the label can be placed relative to the labeled point. Write/Read

Type
Long

Example
```
Dim mValueMapLabelRenderer
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
mValueMapLabelRenderer.RotationalAngles = 30
```

ValueMapLabelRenderer.Symbol property

Description
Symbol for the ValueMapLabelRenderer at the specified index. Write/Read

Syntax
Symbol(Index as long) as Object

Argument
The number you provide for the index specifies the position of the symbol in the collection.

Type
Object

Example
```
Dim mValueMapLabelRenderer
Dim mTextSymbol
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mResult = mValueMapLabelRenderer.Add(mTextSymbol, imsRange.imsExact, 233, 0, 0)
mSymbol = mValueMapLabelRenderer.Symbol(1)
```

ValueMapLabelRenderer.Upper property

Description
Upper value for a specified range in a ValueMapLabelRenderer. Write/Read

Syntax
Upper(Index as Long) as Variant

Argument
The number you provide for the index specifies the position of the value in the collection.

Type
Variant

Example
```
Dim mValueMapLabelRenderer
Dim mTextSymbol
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mResult = mValueMapLabelRenderer.Add(mTextSymbol, imsRange.imsRange, 0, 31, 76)
mUpper = mValueMapLabelRenderer.Upper(1)
```

See Also
ValueMapLabelRenderer.Lower

ValueMapLabelRenderer.Value property

Description
Value for exact symbol by index. Write/Read

Syntax
Value(Index as Long) as Variant

Argument
The number you provide for the index specifies the position of the value in the collection.

Type
Variant

Example
```
Dim mValueMapLabelRenderer
Dim mTextSymbol
Set mValueMapLabelRenderer = Server.CreateObject("aims.ValueMapLabelRenderer")
Set mTextSymbol = Server.CreateObject("aims.TextSymbol")
mResult = mValueMapLabelRenderer.Add(mTextSymbol, imsRange.imsExact, 233, 0, 0)
mValue = mValueMapLabelRenderer.Value(1)
```

ValueMapRenderer.Add method

Description

Adds a symbol to the ValueMapRenderer.

Syntax

ValueMapRenderer.Add(symbol as Object, range as Long, value, lower, upper, Optional Equality as Long [default = 2], Optional Method as Long [default = 1***Optional Label As String***]), Optional Label as String [no default]

Arguments

Symbol	Object	Symbol object to add.
Range	Variant	Type constant of the range (Exact, Range, or Other).
Value	Variant	Variant value for exact symbol.
Lower	Variant	Lower value for range symbol.
Upper	Variant	Upper value for range symbol.
Equality	Long	Optional value for Equality. Used with imsRange.imsRange.
Method	Long	Optional value for Method. Used with imsRange.imsExact.
Label	String	Optional label as String. Label for legend.

Returned Value

Boolean

Example

```
Dim mValueMapRenderer
Dim mSimpleMarkerSymbol
Set mValueMapRenderer = Server.CreateObject("aims.ValueMapRenderer")
Set mSimpleMarkerSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
mSimpleMarkerSymbol.Color = imsColor.imsRed
mResult = mValueMapRenderer.Add(mSimpleMarkerSymbol, imsRange.imsExact, 25, 0,
0,0,1)
mSimpleMarkerSymbol.Color = imsColor.imsBlue
mResult = mValueMapRenderer.Add(mSimpleMarkerSymbol, imsRange.imsRange, 0, 26,
31,2)
mSimpleMarkerSymbol.Color = imsColor.imsGreen
mResult = mValueMapRenderer.Add(mSimpleMarkerSymbol, imsRange.imsOther, 0, 0, 0)
```

See Also

GradientFillSymbol methods and properties
HashLineSymbol methods and properties
RasterFillSymbol methods and properties
RasterMarkerSymbol methods and properties
SimpleLineSymbol methods and properties
SimpleMarkerSymbol methods and properties
SimplePolygonSymbol methods and properties
TrueTypeMarkerSymbol methods and properties

ValueMapRenderer.Clear method

Description
Clears all of the class break symbols and values from the ValueMapRenderer.

Syntax
ValueMapRenderer.Clear()

Arguments
None

Returned Value
None

Example
```
Dim mValueMapRenderer
Dim mSimpleMarkerSymbol
Set mValueMapRenderer = Server.CreateObject("aims.ValueMapRenderer")
Set mSimpleMarkerSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
mResult = mValueMapRenderer.Add(mSimpleMarkerSymbol, imsRange.imsExact, 266, 0, 0)
Set mValueMapRenderer.Clear()
```

ValueMapRenderer.Clone method

Description
Clones the ValueMapRenderer.

Syntax
ValueMapRenderer.Clone()

Arguments
None

Returned Value
ValueMapRenderer Cloned ValueMapRenderer.

Example
```
Dim mClone
Dim mValueMapRenderer
Set mValueMapRenderer = Server.CreateObject("aims.ValueMapRenderer")
Set mClone = mValueMapRenderer.Clone()
```

ValueMapRenderer.Count property

Description
Number of symbols in the ValueMapRenderer. Read only

Type
Long

Example
```
Dim mValueMapRenderer
Dim mSimpleMarkerSymbol
Set mValueMapRenderer = Server.CreateObject("aims.ValueMapRenderer")
Set mSimpleMarkerSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
mResult = mValueMapRenderer.Add(mSimpleMarkerSymbol, imsRange.imsExact, 233, 0, 0)
mCount = mValueMapRenderer.Count
```

ValueMapRenderer.Equality property

Description
Used in conjunction with imsRange. Determines whether or not the bounds of the range (the bounding values) are included in a set or returned range, as follows:

all—all values in the range are returned or set

lower—all values excluding the highest value in the range are returned or set

upper—all values excluding the lowest value in the range are returned or set

Write/read.

Syntax
Equality(Index as Long) as imsEquality

Argument
The number you provide for the index specifies the position of the equality in the collection.

Type
Long

Example
```
Dim mValueMapRenderer
Dim mSimpleMarkerSymbol
Set mValueMapRenderer = Server.CreateObject("aims.ValueMapRenderer")
mValueMapRenderer.LookupField = "COUNT"
Set mSimpleMarkerSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
mResult = mValueMapRenderer.Add(mSimpleMarkerSymbol,imsRange.imsRange,0,0,10000,
imsMethod.imsAll,0)
mMethod = mValueMapRenderer.Equality(1)
```

See Also
imsRange

ValueMapRenderer.Label property

Description
Label for legend. Write/Read

Syntax
Label(Index as Long) as String

Argument
The number you provide for the index specifies the position of the label in the collection.

Type
String

Example
```
Set valuerender=Server.CreateObject("aims.ValueMapRenderer")
valuerender.LookupField="POP1990"
Set sps1=Server.CreateObject("aims.SimplePolygonSymbol")
sps1.FillColor=RGB(255,0,255)
valuerender.Add sps1,imsExact,1453,0,0
valuerender.Label(1)="this is the legend for pop1990=1453"
```

ValueMapRenderer.LookupField property

Description
Name of the field used for specifying ranges for Range or exact values for Exact. Read only

Type
String

Example
```
Dim mValueMapRenderer
Set mValueMapRenderer = Server.CreateObject("aims.ValueMapRenderer")
mValueMapRenderer.LookupField = "CNTRY_NAME"
```

ValueMapRenderer.Lower property

Description
Lower value of a range for a symbol in a ValueMapRenderer. Write/Read

Syntax
Lower(Index as Long) as Variant

Argument
The number you provide for the index specifies the position of the lower value in the collection.

Type
Variant

Example
```
Dim mValueMapRenderer
Dim mSimpleMarkerSymbol
Set mValueMapRenderer = Server.CreateObject("aims.ValueMapRenderer")
Set mSimpleMarkerSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
mResult = mValueMapRenderer.Add(mSimpleMarkerSymbol, imsRange.imsExact, 233, 0, 0)
mLower = mValueMapRenderer.Lower(1)
```

ValueMapRenderer.Method property

Description
Used in conjunction with imsRange.imsExact. Returns or sets the way a value in the data field is compared to the Exact value. Use imsMethod.isExact for an exact match. Use imsMethod.isContained to search for the value anywhere in a string. String comparisons are case sensitive. Write/Read

Syntax
Method(Index as Long) as imsMethod

Argument
The number you provide for the index specifies the position of the method in the collection.

Type
Long

Example
```
Dim mValueMapRenderer
Dim mSimpleMarkerSymbol
Set mValueMapRenderer = Server.CreateObject("aims.ValueMapRenderer")
mValueMapRenderer.LookupField = "COUNT"
Set mSimpleMarkerSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
mResult = mValueMapRenderer.Add(mSimpleMarkerSymbol, imsRange.imsExact,100,0,10,
,imsMethod.imsExact)
mMethod = mValueMapRenderer.Method(1)
```

See Also
imsRange

ValueMapRenderer.Range property

Description

Type of the range (Exact, Range, or Other) for a specified symbol in a ValueMapRenderer. This is the index to a ValueMapRenderer symbol. The possible values are:

0 = imsExact
1 = imsRange
2 = imsOther

Write/Read.

Syntax

Range(Index as Long) as imsRange

Argument

The number you provide for the index specifies the position of the range in the collection.

Type

Long

Example

```
Dim mValueMapRenderer
Dim mSimpleMarkerSymbol
Set mValueMapRenderer = Server.CreateObject("aims.ValueMapRenderer")
Set mSimpleMarkerSymbol = Server.CreateObject("aims.TextSymbol")
mResult = mValueMapRenderer.Add(mSimpleMarkerSymbol, imsRange.imsExact, 233, 0, 0)
mRange = mValueMapRenderer.Range(1)
```

See Also

imsRange

ValueMapRenderer.Remove method

Description

Removes symbols from the ValueMapRenderer by index.

Syntax

ValueMapRenderer.Remove (index as Long)

Arguments

Index Long The index of a symbol in the symbol collection of the ValueMapRenderer.

Returned Value

Boolean

Example

```
Dim mValueMapRenderer
Dim mSimpleMarkerSymbol
Set mValueMapRenderer = Server.CreateObject("aims.ValueMapRenderer")
Set mSimpleMarkerSymbol = Server.CreateObject("aims.TextSymbol")
mResult = mValueMapRenderer.Add(mSimpleMarkerSymbol, imsRange.imsExact, 233, 0, 0)
mResult = mValueMapRenderer.Remove(1)
```

ValueMapRenderer.Symbol property

Description
Symbol for the ValueMapRenderer at the specified index. Write/Read

Syntax
Symbol(Index as Long)

Argument
The number you provide for the index specifies the position of the symbol in the collection.

Type
Object

Example
```
Dim mValueMapRenderer
Dim mSimpleMarkerSymbol
Set mValueMapRenderer = Server.CreateObject("aims.ValueMapRenderer")
Set mSimpleMarkerSymbol = Server.CreateObject("aims.TextSymbol")
mResult = mValueMapRenderer.Add(mSimpleMarkerSymbol, imsRange.imsExact, 233, 0, 0)
mSymbol = mValueMapRenderer.Symbol(1)
```

ValueMapRenderer.Upper property

Description
Upper value for a specified range in a ValueMapRenderer. Write/Read

Syntax
Upper(Index as Long) as Variant

Argument
The number you provide for the index specifies the position of the upper value in the collection.

Type
Variant

Example
```
Dim mValueMapRenderer
Dim mSimpleMarkerSymbol
Set mValueMapRenderer = Server.CreateObject("aims.ValueMapRenderer")
Set mSimpleMarkerSymbol = Server.CreateObject("aims.TextSymbol")
mResult = mValueMapRenderer.Add(mSimpleMarkerSymbol, imsRange.imsRange, 0, 31, 76)
mUpper = mValueMapRenderer.Upper(1)
```

See Also
ValueMapRenderer.Lower

ValueMapRenderer.Value property

Description
Value for an exact symbol by index. Write/Read.

Syntax
Value(Index as Long) as Variant

Argument
The number you provide for the index specifies the position of the symbol in the collection.

Type
Variant

Example
```
Dim mValueMapRenderer
Dim mSimpleMarkerSymbol
Set mValueMapRenderer = Server.CreateObject("aims.ValueMapRenderer")
Set mSimpleMarkerSymbol = Server.CreateObject("aims.SimpleMarkerSymbol")
mSimpleMarkerSymbol.Width = 10
mResult = mValueMapRenderer.Add(mSimpleMarkerSymbol, imsRange.imsExact, 266, 0,
0)
mValue = mValueMapRenderer.Value(1)
```