

# **Customizing ArcIMS™**

**ArcIMS™ 3**

**HTML Viewer**



# **Customizing ArcIMS**

**ArcIMS™ 3**

**HTML Viewer**



Copyright © 2000 Environmental Systems Research Institute, Inc.

All rights reserved.

Printed in the United States of America.

The information contained in this document is the exclusive property of Environmental Systems Research Institute, Inc. This work is protected under United States copyright law and the copyright laws of the given countries of origin and applicable international laws, treaties and/or conventions. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as expressly permitted in writing by Environmental Systems Research Institute, Inc. All requests should be sent to Attention: Contracts Manager, Environmental Systems Research Institute, Inc., 380 New York Street, Redlands, CA 92373-8100, USA.

The information contained in this document is subject to change without notice.

**U.S. GOVERNMENT RESTRICTED/LIMITED RIGHTS**

Any software, documentation, and/or data delivered hereunder is subject to the terms of the License Agreement. In no event shall the U.S. Government acquire greater than RESTRICTED/LIMITED RIGHTS. At a minimum, use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR §52.227-14 Alternates I, II, and III (JUN 1987); FAR §52.227-19 (JUN 1987) and/or FAR §12.211/12.212 (Commercial Technical Data/Computer Software); and DFARS §252.227-7015 (NOV 1995) (Technical Data) and/or DFARS §227.7202 (Computer Software), as applicable. Contractor/Manufacturer is Environmental Systems Research Institute, Inc., 380 New York Street, Redlands, CA 92373-8100, USA.

ESRI and the ESRI globe logo are trademarks of Environmental Systems Research Institute, Inc., registered in the United States and certain other countries; registration is pending in the European Community. ArcGIS, ArcIMS, ArcSDE, ArcExplorer, GIS by ESRI, and the ArcIMS logo are trademarks and www.esri.com and @esri.com are service marks of Environmental Systems Research Institute, Inc. Netscape and the Netscape N logo are registered trademarks of Netscape Communications Corporation in the United States and other countries. Microsoft and the Windows logo are registered trademarks and the Microsoft Internet Explorer logo is a trademark of Microsoft Corporation. Other companies and products mentioned herein are trademarks or registered trademarks of their respective trademark owners.

# Contents

## 1 Introducing the HTML Viewer 5

- What is the HTML Viewer? 6
- HTML Viewer file organization 7
- HTML Viewer frames 8

## 2 Customizing the HTML Viewer 11

- Working with ArcIMSParam.js 12
- Working with MapServices and map layout 13
- Working with an acetate layer 15
- Modifying attribute data display 16
- Changing the title, logo, and colors 17
- Working with tools and the toolbar 18
- Creating hyperlinks 20
- How the viewer and server communicate 21
- JavaScript function files 27
- Using the sample HTML Viewers 28

## 3 HTML Viewer JavaScript Library 34

- Organization of the HTML Viewer JavaScript Library 35
- JavaScript functions by name (list) 38
- JavaScript functions by category (list) 44
- Function descriptions 50
- Global variables by name (list) 234
- Global variables by category (list) 245
- aimsBuffer.js 257
- aimsClick.js 258
- aimsGeocode.js 260
- aimsIdentify.js 262
- aimsLayers.js 263
- aimsMap.js 267
- aimsPrint.js 272
- aimsQuery.js 273
- aimsSelect.js 274
- aimsXML.js 276
- ArcIMSParam.js 278
- MapFrame.htm 289

# Introducing the HTML Viewer

# 1

## IN THIS CHAPTER

- **What is the HTML Viewer?**
- **HTML Viewer file organization**
- **HTML Viewer frames**

ESRI® ArcIMS™ 3 software provides a suite of tools allowing you to create very effective web sites for your mapping and GIS needs. The ArcIMS Viewers provide the foundation for the graphical and functional components of these web sites. You can build on this foundation through customization of the ArcIMS Viewers.

*Customizing ArcIMS* is a series of books that describes the customization of the ArcIMS Viewers and their programming references. This series covers customization using the HTML and Java™ Viewers and the ActiveX® and ColdFusion® Connectors.

This book explains the foundation for customizing the ArcIMS HTML client, or viewer, as it is commonly referred to, as well as provides a complete reference to the function library available with ArcIMS.

This book assumes that you have a working knowledge of HTML and familiarity with JavaScript™.

In this chapter, you are introduced to

- Reasons for customizing the HTML Viewer
- The file structure and frame layout of the HTML Viewer
- The relationship between key HTML and JavaScript files with the HTML Viewer

## What is the HTML Viewer?

If you use ArcIMS Designer to create web sites, you are probably already familiar with the HTML Viewer. The HTML Viewer defines the graphical look and functionality of your ArcIMS web sites. The default HTML Viewer is a set of HTML pages and JavaScript files that reflect the choices you make in the panels of ArcIMS Designer.

The HTML Viewer provides a framework for the map, toolbar, legend, overview map, and other graphic portions of the web site. Starting with this initial framework, you can quickly customize the web site.

Even with the many choices you have in ArcIMS Designer, you may still need to be more flexible in order to implement a more customized look in the design of your web site.

ArcIMS Designer creates output files, including HTML and JavaScript files (.js), that form the foundation of the HTML Viewer. HTML files are used to generate web page content, and the JavaScript functions allow for user interaction with the map.

You may want to customize in the following ways:

- Changing the frame layout
- Modifying the toolbar
- Adding functionality
- Changing the graphic look
- Inserting your own company logo

### Considerations for choosing the HTML Viewer

ArcIMS provides four customizable clients—HTML, Java, ActiveX, and ColdFusion.

The HTML Viewer can be easily and extensively customized. Along with HTML, the HTML Viewer also incorporates a significant amount of JavaScript and some Dynamic HTML (DHTML). These technologies are increasingly being used by web developers to add flexibility and interactivity to their web pages.

The HTML Viewer performs less processing on the client machine than the Java Viewer—this is often referred to as a “lighter” or “thinner” client. ActiveX and ColdFusion are the thinnest clients.

HTML is the most widely accepted and supported language on the Web for defining page content. It does not require the Java 2 plug-in or Applet support. Its reliance on JavaScript, however, makes it behave differently in different browsers.

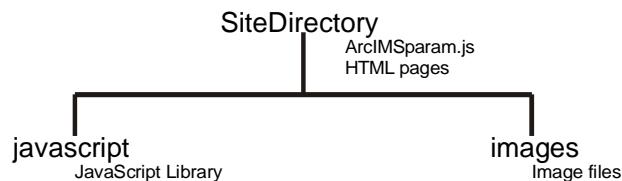
The HTML Viewer is the best solution when building a web site that incorporates one Image MapService. If you want to include more than one Image MapService or one or more Feature MapServices in your web site, you need to work with the Java Viewer.

The HTML Viewer is also the viewer of choice for web sites that don't include Feature MapServices or require users to add their own data to the map. The HTML Viewer does not support some functionality used by Feature MapServices such as EditNotes, MapTips, and MapNotes. Data integration, such as users adding local data layers, is not supported by the HTML Viewer. For a complete matrix of functionality supported in both viewers, see *Using ArcIMS*, Chapter 4, ‘Designing a Web Site’, ‘Viewer Functionality’.

# HTML Viewer file organization

## Directory structure

When you create a web site using the default HTML Viewer through ArcIMS Designer, a hierarchy of directories and files is created. The web site directory contains a set of HTML files and a parameters file, along with two subdirectories—javascript and images.



The javascript subdirectory contains the files that make up the HTML Viewer JavaScript Library. These JavaScript files contain the functions that perform many of the common operations for the HTML Viewer. For an overview of the JavaScript files, see Chapter 2, ‘Customizing the HTML Viewer’, and for a detailed functional reference, see Chapter 3, ‘HTML Viewer JavaScript Library’.

The images subdirectory contains the graphic images used in the viewer pages such as the buttons, icons, and logos. When building a new site, you may consider replacing these images to create your own corporate or departmental look.

You may notice another subdirectory named Meta-inf. This subdirectory is created when building a standard viewer from Designer, but is not needed for any customization of the viewer. You can delete this directory to make the HTML Viewer lighter.

## The ArcIMSpParam.js parameters file

Located in the web site directory is a parameters file named ArcIMSpParam.js. This file contains a set of JavaScript variables that affect the web site’s look and behavior. A significant amount of customization can be done by changing the variables in this file to suit your preferences. Some of the more commonly modified items in this file are tool display and operation, layer management, and colors and graphics of the site. See Chapter 2, ‘Customizing the HTML Viewer’, for more information on customizing the ArcIMSpParam.js file.

## The HTML files

There are approximately 20 HTML files that define the page content for the HTML Viewer. Default.htm starts the process and viewer.htm defines the structure of the HTML Viewer.

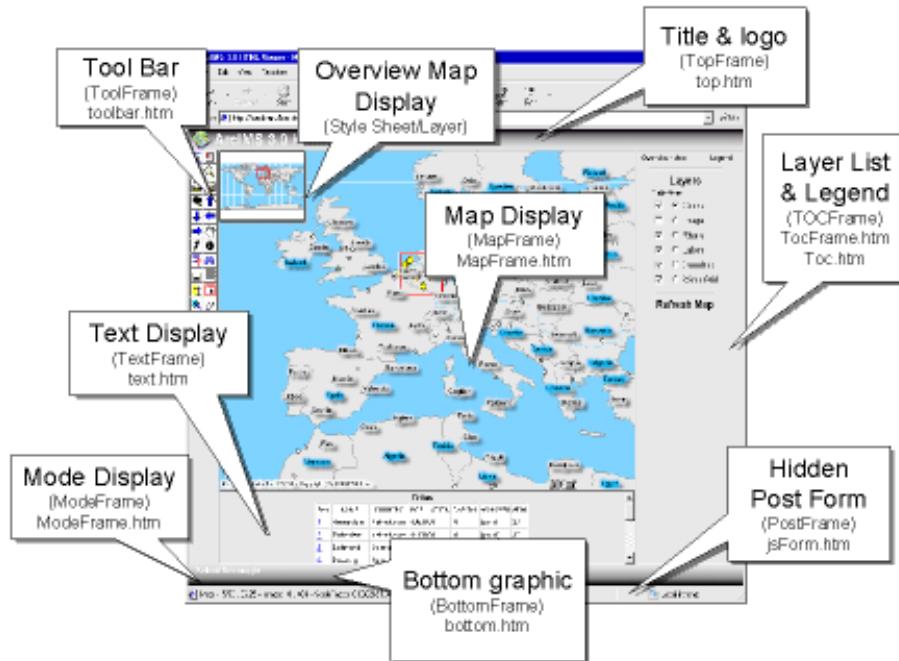
Default.htm is the entry point into your ArcIMS web site. Index.htm redirects to Default.htm and is used by browsers that have index.htm as their initial page. Default.htm handles the following four tasks—creates page title, checks which browser is being used, establishes URL parsing rules, and writes out a file that defines the frames for the page layout named viewer.htm.

Viewer.htm is the file that contains the frame layout for the HTML Viewer. The frames provide structure, rather than content, to the layout by dividing up the web browser space into multiple “sections”. Each of the sections gets its content from an HTML file.

Other HTML files that provide frame content are listed later in this chapter under ‘HTML Viewer frame layout’.

## HTML Viewer frames

Familiarity with HTML frames is important in understanding the relationship between the files that make up the web site. Each frame displays an HTML page that works in coordination with the pages in the other frames. The HTML Viewer created by ArcIMS Designer is shown below.



### TopFrame

Top.htm defines the content for TopFrame. TopFrame is across the top of the viewer and contains the title text and ArcIMS logo.

### MapFrame and overview map

MapFrame.htm defines the content for MapFrame. MapFrame displays the map image. When the frame loads, it sets a variety of additional map parameters and loads additional JavaScript files to enable the viewer to perform much of its functionality. See a complete listing of all JavaScript files in Chapter 3, ‘HTML Viewer JavaScript Library’.

The overview map is included in the MapFrame. It is defined in MapFrame.htm using a dynamic cascading style sheet in Internet Explorer or a layer in Netscape. The OverView Map tool on the toolbar toggles the visibility.

The scale bar that appears on the map is an element on the acetate layer that is “overlaid” on the map. See Chapter 2, ‘Customizing the HTML Viewer’, for information on using the acetate layer.

## **ToolFrame**

Toolbar.htm defines the content for ToolFrame. ToolFrame contains a panel of buttons (toolbar) used to select the current Viewer tool. Toolbar.htm checks the values of key parameters set in ArcIMSParam.js, and using JavaScript, it dynamically creates the toolbar based on the tool selected.

## **TOCFrame**

TOCFrame.htm and toc.htm define the content for TOCFrame. TOCFrame.htm is loaded initially, then toc.htm is written to define the LayerList and Legend displayed in TOCFrame. When displaying the LayerList, toc.htm lists only the layers visible at the current scale. When the scale changes, for example when you zoom in, toc.htm is reloaded.

## **ModeFrame**

ModeFrame.htm defines the content for ModeFrame. ModeFrame displays the current cursor mode, which is defined by the current tool. When a different tool is clicked, the mode changes and ModeFrame.htm is reloaded.

## **TextFrame**

Text.htm is a placeholder for the content of TextFrame. The content for TextFrame is dynamically written based on the tool chosen. It displays forms and information returned from the ArcIMS Spatial Server. For example, TextFrame displays the results for the Identify tool or the form for the Query tool. Other HTML files, such as findForm.htm, addmatch.htm, setUnits.htm, and select.htm, fill this frame when the associated tool is selected.

## **PostFrame**

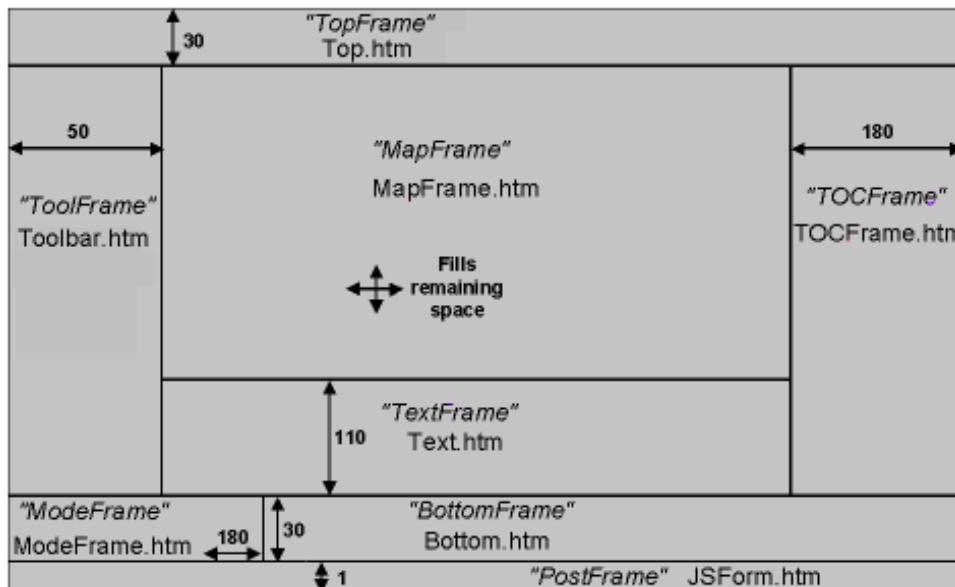
JsForm.htm defines the content for PostFrame. PostFrame is a hidden frame that contains the hidden form named Post Form. This form sends the ArcXML requests to the ArcIMS Servlet Connector. See Chapter 2, ‘Customizing the HTML Viewer’, ‘How the viewer and server communicate’, for more information on ArcXML requests and responses.

## **BottomFrame**

Bottom.htm defines the content for BottomFrame. BottomFrame is positioned next to the ModeFrame to visually complete the graphic along the bottom of the viewer.

## HTML Viewer frame layout

The diagram below shows the default layout for the Internet Explorer HTML Viewer (Netscape is slightly different). Each area shows the name of the frame, the name of the HTML file that fills it, and its size in pixels. The diagram is not to scale, but provides a guideline for planning the layout for a new custom viewer.



A given web site may be broken into many rows and columns—as is the case in the example above—or it can have just one or two frames. Determining the location and size of frames is completely up to you. First determine the frames you want in the web site, then allocate the necessary space.

# Customizing the HTML Viewer

# 2

## IN THIS CHAPTER

- **Working with ArcIMSParam.js**
- **Working with MapServices and map layout**
- **Working with an acetate layer**
- **Modifying attribute data display**
- **Changing the title, logo, and colors**
- **Working with tools and the toolbar**
- **Creating hyperlinks**
- **How the viewer and the server communicate**
- **JavaScript function files**
- **Using the sample HTML Viewers**

If you have used ArcIMS even a few times, you are probably already familiar with the HTML Viewer that is created from ArcIMS Designer. This default HTML Viewer can be greatly customized to create the look you want for your web site.

To customize the HTML Viewer you primarily work with the ArcIMSParam.js parameter file and a set of JavaScript functions. Even if you are new to JavaScript, you can still accomplish a significant amount of customization by just altering the ArcIMSParam.js file.

In this chapter, you learn how to

- Complete common customization tasks.
- Follow the communication between the viewer and the server.
- Use the HTML Viewer samples as examples for customization.

## Working with ArcIMSpParam.js

When you are ready to create a custom web site, the ArcIMSpParam.js file is likely the first file you'll modify. ArcIMSpParam.js is found in the web site directory and is well documented. It contains global variables defining both the look and behavior of the web site, and you can do a significant amount of customization by simply modifying these variables. Descriptions of the variables are found in Chapter 3, ‘The HTML Viewer JavaScript Library’.

All variables can be altered either directly or through dynamic HTML (DHTML) to produce a custom look. ArcIMSpParam.js includes, but is not limited to, variables for

- Changing MapServices and map extents
- Setting the initial display of the legend
- Defining North arrows and copyright text on an acetate layer
- Defining fields for attribute display
- Changing tools and the toolbar

The following pages show you how to modify the ArcIMSpParam.js file to make some of these changes to the viewer. The variable names are shown in italics. All of the changes are done in ArcIMSpParam.js unless otherwise noted.

### **CheckParams function**

ArcIMSpParam.js contains a function called `checkParams`, which is called on startup after the viewer has loaded all the files. This function checks for the existence of the various frames and function files, checks the value of key variables, and updates many parameters.

### **Refreshing your web site to reflect changes**

When you make a change to the variables in JavaScript or HTML files and want to view the results, you may need to stop and start your web server, close the browser and clear the Internet cache, or just refresh the site. It depends on the combination of web server and web browser you are using and their configuration. This causes the files to be reread and any changes get incorporated into the web site.

# Working with MapServices and map layout

## Changing the MapService

Set the variable *imsURL* to the URL of the MapService used in the main map display.

## Changing the Overview MapService

Set the variable *imsOVURL* to the URL of the MapService used in the overview map display.

## Removing the Overview Map

Set the variable *hasOVMap* to false in ArcIMSPparams.js and set *ovIsVisible* to false in MapFrame.htm. The first change removes the tool from the toolbar and disables requests for the map. The second change makes the style sheet that creates the overview map invisible.

## Changing the starting map extent

Set the values of these four variables to the new coordinates: *startLeft*, *startBottom*, *startRight*, and *startTop*. This sets Minimum X, Minimum Y, Maximum X, and Maximum Y, respectively.

## Displaying the graphic legend at startup instead of the LayerList

The LayerList shows the layers with a checkbox and radio button for each layer. The checkbox indicates visibility and the radio button indicates whether its active. The legend is an image showing layers and their symbology.

Set the variable *showTOC* to false, and add the variable *legendVisible* and set it to true in the function *checkParams* as shown below.

```
function checkParams() {  
    legendVisible=true;  
    appDir=getPath(document.location.pathname);
```

The variable *legendVisible* is initially set to false in aimsMap.js. You can also go to this file and change the value to true to obtain the same result.

## Setting an active layer

When the LayerList is removed from the viewer, the user does not have the ability to set the active layer. The active layer is required by some tools, such as Identify. Removing the LayerList reclaims space for the map and can simplify the interface for the user of the site, but you must handle setting the active layer.

To set an active layer, in the ArcIMSPparam.js file set the variable *ActiveLayerIndex* to the index number of the layer you want active. An index number of 0 indicates the first (topmost) layer, 1 the second, and so on. The default value of *ActiveLayerIndex* is 99. If the index number is higher than the number of layers, the first layer is made active.

The order of the layers in the legend is dependent on the order of the layers in the Map Configuration file (.axl). In this example, the Customers layer is being set as the active layer. In the Map Configuration file, the <LAYER> tags are in this order—zip, trade80, streets, customers, stores. The index numbers are 4, 3, 2, 1, and 0, respectively. When the layers are placed in the legend, they go in reverse, so zip is on the bottom at index 4, and the stores are on top at index 0. Given this, to make customers active, set

```
var ActiveLayerIndex=1;
```

### Replacing the animated graphics

The HTML Viewer uses a couple of animated graphics to give user feedback when it's busy retrieving map images and data. The Retrieving Map and Retrieving Data graphics are referenced in MapFrame.htm. You can create animated graphics in a variety of graphics programs, but you can also replace them with a static graphic that appears and disappears when appropriate.

The following code from MapFrame.htm provides the references to these graphics. The graphics reside in the web site /images directory. When you replace these graphics you must alter the height and width parameters to match the height and width of your graphics.

```
// loading splash  
content = '';  
createLayer("LoadData",loadBannerLeft,loadBannerTop,273,30,false,content);  
content = '';  
createLayer("LoadMap",loadBannerLeft,loadBannerTop,273,30,false,content);
```

Immediately following the <BODY> tag, you will find a set of variables. You also need to change the height and width in the variables *loadBannerLeft* and *loadBannerTop*. These variables are used to center the graphic on the page.

```
var mwidth = getMapwidth();  
var mheight = getMapHeight();  
var loadBannerLeft = parseInt((mwidth - 273)/2);  
var loadBannerTop = parseInt((mheight - 30)/2);
```

# Working with an acetate layer

An acetate layer can be thought of as a clear piece of transparent film that sits on top of the map. An acetate layer displays auxiliary information on the map. The HTML Viewer contains several acetate layers that display a North arrow, copyright text, and scale bar. You can change these elements or add new elements.

## Changing the properties of the North arrow

The North arrow is displayed if *drawNorthArrow* is true. To modify the type, position, and size, use the variables *NorthArrowType*, *NorthArrowCoords*, *NorthArrowSize*, and *NorthArrowAngle*.

## Changing the properties of the copyright element

The copyright is displayed on the map if *drawCopyright* is true. To modify the text, font, size, and so on, use the variables *CopyrightFont*, *CopyrightStyle*, *CopyrightSize*, *CopyrightCoords*, *CopyrightColor*, *CopyrightBackground*, *CopyrightBGColor*, *CopyrightGlow*, *CopyrightGlowColor*, and *CopyrightText*.

## Changing the properties of the scale bar

The scale bar is displayed if *drawScaleBar* is true. To modify the style, font, size, color, and so on, use the variables *ScaleBarStyle*, *ScaleBarFont*, *ScaleBarFontSize*, *ScaleBarSize*, *ScaleBarColor*, *ScaleBarBackground*, and *ScaleBarBackColor*.

## Adding a new element to an acetate layer

The acetate layers are created in the aimsXML.js file. This example shows the acetate layer for the North arrow.

```
theString += '<LAYER type="ACETATE" name="theNorthArrow">\n';
theString += '<OBJECT units="PIXEL">\n<NORTHARROW type="" + NorthArrowType +
" size="" + NorthArrowSize + " coord="" + NorthArrowCoords + "
shadow="32,32,32" ';
theString += 'angle="" + NorthArrowAngle + " antialiasing="True"
overlap="False" />\n</OBJECT>\n';
theString += '</LAYER>\n';
```

If you want to add your own elements to an acetate layer, open the aimscustom.js file and add syntax to the *addCustomToMap1*, *addCustomToMap2*, *addCustomToMap3*, or *addCustomToMap4* function. The difference between them is the stacking order of the acetate layer. An example is shown below.

```
function addCustomToMap3(){
    var customString = "";
    customString += '<LAYER type="ACETATE" name="MyCompanyText">\n';
    customString += '<OBJECT units="PIXEL">\n<TEXT coord="135,' + (iHeight-20) +
    '"label="Company X rocks!">\n';
    customString += '<TEXTMARKERSYMBOL fontstyle="BOLD" fontsize="12"
    font="ARIAL" fontcolor="" + modeMapColor + " " ;
    customString += 'threed="TRUE" glowing="" + modeMapGlow + " />\n</TEXT>
    \n</OBJECT>\n</LAYER>\n';
```

# Modifying attribute data display

## Limiting the fields displayed

To limit the fields returned in a selection, query, or identify, change the value of the variable *selectFields* to set the fields you want displayed. The default value is #ALL#, which indicates all fields are displayed.

Since query operations are typically done on the active layer, you probably want the field display to change when the active layer changes. To make this happen, set *swapSelectFields* to true. If *swapSelectFields* is true, then a list of field names must be created for each layer.

To create the list of fields for a layer, set the array variable *selFieldList*. Assign an element for each layer in this array, with the topmost layer set at index 0. Each line of the array is assigned like this:

```
selFieldList[2] = "NAME #ID# #SHAPE# POP";
```

An element is required for each layer. The ID and Shape fields must be included in the list and must be surrounded by #s. This notation indicates that these fields are not in the database but instead are generated by the server. Image layers are assigned #ALL# since they have no attributes.

An example of the assignment of these three variables is shown in Chapter 3, ‘The HTML Viewer JavaScript Library’, ‘ArcIMSParams.js’, ‘Identify>Select/Query/Buffer parameters’.

## Using aliases for the field names

To display an alias field name instead of its original name, set *useFieldAlias* to true. When *useFieldAlias* is true, a list of field names and their aliases must be created for each layer.

To create the list of field names and aliases for a layer, set the array variable *fieldAliasList*. Assign an element for each layer in this array, with the topmost layer set at index 0. The list is a string containing pairs of field names and their aliases, separated by a colon. Each pair is separated by a bar (“|”). Each element of the array is assigned like this:

```
fieldAliasList [0] = "NAME:City Name | POP:Population";
fieldAliasList [1] = "";
```

Because an element is required for each layer, if you don’t want to assign aliases for a layer, set the element list to an empty string (“”) as shown for element [1]. The viewer checks for an alias to use and only swaps the field name if it finds a name/alias pair for that layer’s field in the list.

An example of this assignment is shown in the description for the *useFieldAlias* array in Chapter 3, ‘The HTML Viewer JavaScript Library’, ‘ArcIMSParams.js’, ‘Identify>Select/Query/Buffer parameters’.

## Changing the number of records listed at one time

Although the map displays all selected features, you can control the number of records listed at one time. This is set in the variable *maxFeaturesReturned*. The default value of *maxFeaturesReturned* is set to 25 records.

Changing the value to a larger number impacts response speed and may overload browser capabilities. Links marked More Records and Previous Records are created if the selection count is greater than the *maxFeaturesReturned* variable. The user can use these links to move through the entire selection set.

# Changing the title, logo, and colors

## Changing the title

The title of the HTML Viewer can be set when creating the web site with ArcIMS Designer. The default title is ‘ArcIMS 3.0 Viewer’.

You can change the title after the HTML Viewer is created by editing the default.htm file. Change the following line to include your own title text string:

```
var theTitle = "My Very Own Viewer";
```

## Changing the logo and background on the topFrame

The logo, which appears in the top-left corner of the HTML Viewer, can be changed by editing top.htm. By default, the logo uses the aimslogo1x2.gif from your web site images directory. Edit the location and/or name of the image to change the logo that appears in the topFrame.

The background for the topFrame uses grad\_gray.jpg in the images directory. You can edit the location and/or name of the image to change the background appearance of the topFrame. You should also change this graphic in ModeFrame.htm and bottom.htm to match.

## Changing the color of the box used for zoom area

Set the value of *zoomBoxColor* to a string representing a hexadecimal color or color name. The default is red and this example changes the color to blue.

```
var zoomBoxColor = '#0000ff';
```

Colors in ArcIMS functions and variables are expressed in one of three color models—Hexadecimal, Red:Green:Blue (RGB), or by name. The three color models are not interchangeable. Each function requires a specific color model and you must set the color as defined in the function reference. ArcIMSParam.js provides a default color in the format required.

# Working with tools and the toolbar

## Removing tools from the toolbar

ArcIMSParam.js contains a set of variables that start with ‘use’ representing each tool available in ArcIMS. For example, there are *usePan*, *useZoomIn*, and *useIdentify* variables. The toolbar is created dynamically based on the values of the ‘use’ variables. Set the value of these variables to true if you want to include the tool on the toolbar and false if you don’t want it on the toolbar.

```
//web site path  
var usePan=true;  
var usePanNorth=false;  
var usePanWest=false;  
var usePanEast=false;  
var usePanSouth=false;
```

## Changing search tolerance for Identify

Set the value of *pixelTolerance* to change the number of pixels used for the search tolerance during an Identify operation. Search tolerance is the area created around the click point. The default is 2.

```
var pixelTolerance=5;
```

## Changing the pan and zoom factors

Set the value of the variables *panFactor* and *zoomFactor* to modify the scale factor that occurs when the map is panned or zoomed from a single point.

```
//panning factor for arrow buttons  
var panFactor = 0.85;  
//zoom factors for v.3  
var zoomFactor = 2;
```

## Changing the toolbar images and structure

Toolbar.htm is the file that defines the structure of the toolbar. It uses DHTML to create a two-column table to house the tools selected for the site. You can customize the file to produce a different table layout, use custom images, or not use tables at all.

To incorporate custom images, replace images in the images directory with GIF images you create. If you decide to have a “selected” and “unselected” tool graphic, as the default viewer has, a naming convention has already been established and your image names should adhere to this convention—the name of the tool followed by \_1 is used when the tool is not selected, while the name of the tool followed by \_2 is used when the tool is selected. This occurs only for tools that stay active until they are changed by the user. This is established with the function *setToolPic*. The function *revertToolPic* sets all tools to the unselected state.

To change the table structure of the toolbar, for example, to make a one-column toolbar, you modify the body of toolbar.htm. This contains a dynamically written page describing the toolbar. Here is a sample of the code that writes the two columns:

```
if (parent.MapFrame.useZoomIn) {  
// Zoom In . . . requires aimsNavigation.js  
document.write('<td align="center" valign="middle">');  
document.write('');  
isSecond = !isSecond;  
document.writeln('</td>');  
if (isSecond) document.write('</tr><tr>');
```

To change this to one column, change the last four lines as follows:

```
//isSecond = !isSecond;  
document.writeln('</td>');  
//if (isSecond);  
document.write('</tr><tr>');
```

To make a simple toolbar with no tables, JavaScript, or dynamically written pages, look at the Tracker or Thematic Map sample applications for examples. In most cases, your toolbar will be static, but you may want to include JavaScript for rollovers. See ‘Using the sample HTML Viewers’ later in this chapter.

## Creating hyperlinks

The nature of the Web is to allow people to link to a variety of places. The instructions below, along with the sample application named HyperLink, show you how to create a web site that hyperlinks from a feature on the map or from an attribute display.

For a hyperlink to work, set the variable *useHyperLink* to true. The database for the layer you want to link must contain a field with valid URLs, for example “<http://www.esri.com>”. You also need to create arrays of the layers and field names on which the hyperlink is performed.

To create the list of layers, set the array variable *hyperLinkLayers*. Assign a string for each layer in this array, with the topmost layer set at index 0. The layer names are case-sensitive and should be specified as you see them in the Map Configuration file. To create the list of hyperlink fields, set the array variable *hyperLinkFields*. The field names are case-sensitive and should be specified as you see them in an attribute table listing in the viewer. The associated layer and the field share the same array index. In the example below, both the museums and art galleries contain a field named website that contains a valid URL.

```
hyperLinkLayers[0] = "Museums";
hyperLinkFields[0] = "WEBSITE";
hyperLinkLayers[1] = "Art Galleries";
hyperLinkFields[1] = "WEBSITE";
```

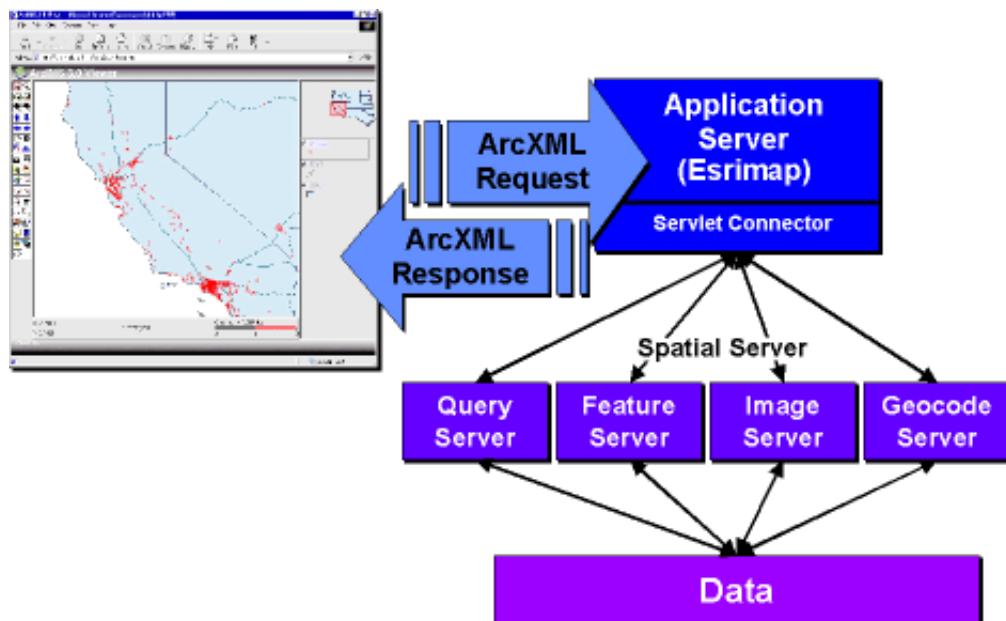
An example of this assignment is shown in the description for these arrays in Chapter 3, ‘The HTML Viewer JavaScript Library’. You can also look at the ‘Using the sample HTML Viewers’ section later in this chapter.

## How the viewer and server communicate

From the user's perspective, a button is clicked, an operation is performed, and a result appears on the screen. This summarizes a very complex process of communication between the viewer and the server.

What actually occurs when the user clicks a button is that an ArcXML request is sent from the viewer to the Application Server. The Application Server directs the request to the appropriate ArcIMS Spatial Server and then returns the response to the viewer.

For an explanation and diagram of the ArcIMS architecture, including the Application Server and Spatial Servers, see *Using ArcIMS*, Chapter 1, 'Introduction to ArcIMS'.

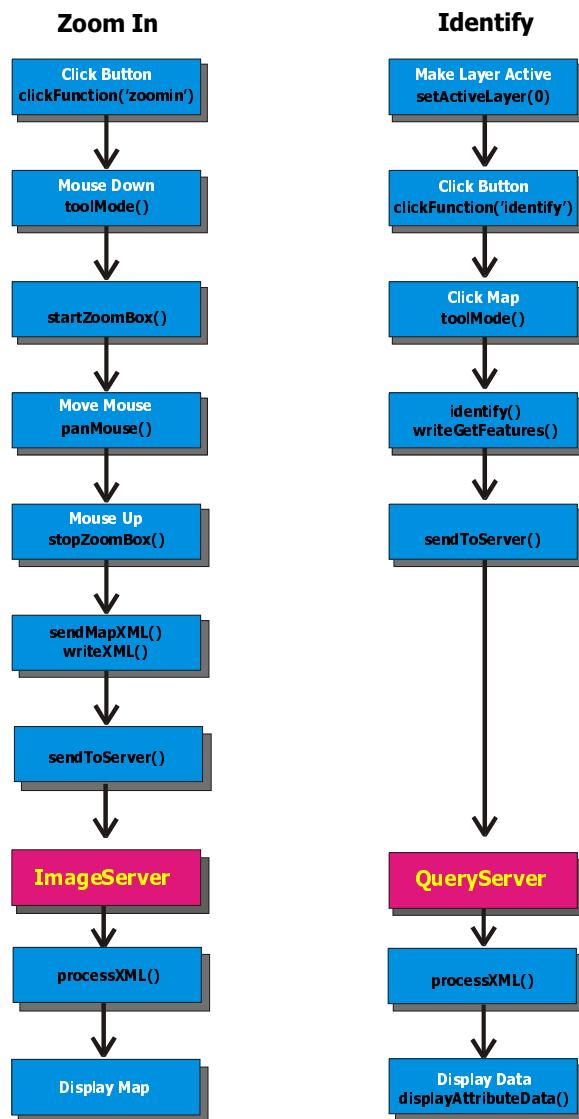


The process described above is still a bit generalized. In order to understand the complete process, there are two more parts to add to the diagram—where the JavaScript functions are called and what exactly happens during the request and response cycle.

## What happens when the mouse is clicked?

This diagram shows an example of the process from the mouse click to the map display for the Zoom In and Identify operations. It includes the JavaScript function calls at each step, when a request is written, when a request is sent to the server (sendToServer), and when it's returned from the server (processXML). The next page describes details for the flow between the sendToServer and processXML functions.

### Examples of Function Flow in the HTML Viewer



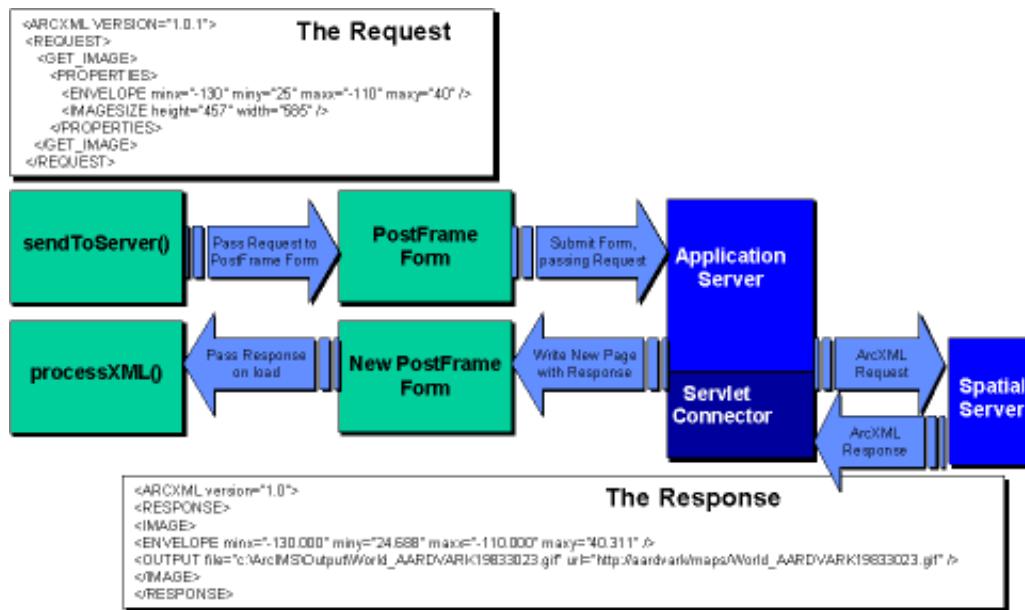
## The ArcXML Request/Response Cycle

Communication between the viewer and the server is based on requests and responses written in ArcXML. The diagram below shows the sequence of events from the client writing an ArcXML request through the server processing the request and sending back the response.

For each cycle, the request updates the form values and passes values to the Application Server Connector, a new page is written with the response, and the new page is used to start the next request. This method of posting to a page is necessary because, by themselves, JavaScript and HTML cannot handle the request and response cycle.

### The Request

For the selected operation, a function writes the appropriate ArcXML request to send to the server. The `sendToServer` function in the MapFrame page then passes the request to the PostFrame page, where it updates a form input value in the PostFrame form. The form is submitted to the Application Server, which redirects it to the Servlet Connector. The Servlet Connector extracts the submitted values and the request is streamed on to the appropriate ArcIMS Spatial Server.



### The Response

The ArcXML response from the ArcIMS Spatial Server is sent back to the Servlet Connector. A new page is dynamically written back to the PostFrame form and replaces the previous page. This new page contains a JavaScript function named `passXML`. It passes the response to another JavaScript function called `processXML` in the MapFrame page. The `processXML` function then passes the ArcXML response to the appropriate function for processing.

## How the PostFrame form works

The PostFrame page originates as jsForm.htm. This file contains a placeholder for the response, and a form containing elements for the XML request, the header and footer file, and the JavaScript function that is called.

Below is an example of the original jsForm.htm file.

```
<HEAD>
  <TITLE>Default Form</TITLE>
  <!-- Title must match header.htm's title -->
  <SCRIPT TYPE="text/javascript" LANGUAGE="JavaScript">
    function passXML() {
      // Esrimap connector writes necessary lines here
    }
  </SCRIPT>
</HEAD>
<BODY BGCOLOR="Black" onload="passXML()">
  <FORM ACTION="" METHOD="POST" name="theForm">
    <!-- <input type="Hidden" name="Form" value="True"> -->
    <INPUT TYPE="Hidden" NAME="ArcXMLRequest" VALUE="">
    <INPUT TYPE="Hidden" NAME="JavaScriptFunction"
    VALUE="parent.MapFrame.processXML">
    <INPUT TYPE="Hidden" NAME="HeaderFile" VALUE="c:/inetpub/wwwroot/htmlviewer/
    header.htm">
    <INPUT TYPE="Hidden" NAME="FooterFile" VALUE="c:/inetpub/wwwroot/htmlviewer/
    footer.htm">
    <INPUT TYPE="Hidden" NAME="FormExists" VALUE="Yes">
  </FORM>
</BODY>
</HTML>
```

During the request, sendToServer( ) updates the ACTION, the value of the element ArcXMLRequest, and the value of the elements HeaderFile and FooterFile. SendToServer( ) then passes these values to the Application Server and Servlet Connector.

The header and footer file locations are stored in the variable *formFilePath* set in ArcIMSParam.js. From this variable, the variables *headerFilePath* and *footerFilePath* are created for each file location.

```
// file locations for servlet connector form creation
var headerFilePath = formFilePath + "/header.htm";
var footerFilePath = formFilePath + "/footer.htm";
```

The web server must be able to understand and resolve the path for these files. For example, if you have a distributed system, any mounted drives used in these pathnames must be accessible from the web server.

During the response, the PostFrame page is replaced. The Servlet Connector brings together three pieces to create the new page—the original header.htm (from *headerFilePath*), the new response, and the original footer.htm (from *footerFilePath*).

- 1) The content of header.htm is written to the top portion of the new page and ends with the function call to passXML.
- 2) This is followed by the response from the Spatial Server in the function passXML which is passed as the variable *XMLResponse*. The last line of the response calls processXML( ) with *XMLResponse*. ProcessXML formats the response into standard ArcXML syntax. It replaces each ‘+’ with a space and replaces all the escape characters. For example, its takes ‘%3c%3fxml+ version’, converts the hexadecimal values 3c to 60 and 3f to 63, then finds the ascii value for each, ‘<’ and ‘?’, respectively. This yields the string ‘<?xml version’ which is the beginning of all ArcXML requests.
- 3) The content from footer.htm is written after the response. It closes the function call, then defines the body and the post form.

Below is an example of the new PostFrame file.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
    <TITLE>Default Form</TITLE>
    <!-- Title must match jsForm.htm's title -->
    <SCRIPT TYPE="text/javascript" LANGUAGE="JavaScript">
        function passXML() {
var XMLResponse='%3c%3fxml+version%3d%221.0%22%3f%3e+%3cARCXML+version%3d%221.0
%22%3e+%3cRESPONSE%3e+%3cIMAGE%3e+%3cENVELOPE+minx%3d%22-122.706100
33333340%22+miny%3d%2237.66946800000002%22+maxx%3d%22-122.11199366666
6660%22+maxy%3d%2237.84770000000003%22+%2f%3e+%3cOUTPUT+file%3d%22D%3a
%5cArcIMS%5coutput%5csanfran_MAMMOTHMTN4303892.jpg%22+url%3d%22http%3a%
2f%2fmammothmtn%2foutput%2fsanfran_MAMMOTHMTN4303892.jpg%22+%2f%3e+%3c%
2fIMAGE%3e+%3c%2fRESPONSE%3e+%3c%2fARCXML%3e+' ;
parent.MapFrame.processXML(XMLResponse);
    }
</SCRIPT>
</HEAD>

<BODY> continued on next page...
```

```
<BODY BGCOLOR="Black" onLoad="passXML()">
<FORM ACTION="" METHOD="POST" name="theForm">
  <!-- <input type="Hidden" name="Form" value="True"> -->
  <INPUT TYPE="Hidden" NAME="ArcXMLRequest" VALUE="">
  <INPUT TYPE="Hidden" NAME="JavaScriptFunction"
  VALUE="parent.MapFrame.processXML">
  <INPUT TYPE="Hidden" NAME="HeaderFile" VALUE="c:/inetpub/wwwroot/htmlviewer/
  header.htm">
  <INPUT TYPE="Hidden" NAME="FooterFile" VALUE="c:/inetpub/wwwroot/htmlviewer/
  footer.htm">
  <INPUT TYPE="Hidden" NAME="FormExists" VALUE="Yes">
</FORM>
</BODY>
</HTML>
```

## JavaScript function files

The following JavaScript function files are included in the HTML Viewer. The ArcIMSPparam.js is located in each web site directory. All other files are located in the web site \javascript directory. They are briefly introduced here and are discussed in more detail in Chapter 3, ‘The HTML Viewer JavaScript Library’.

- ArcIMSPparam.js—main parameter file that configures the HTML Viewer
- AimsMap.js—basic mapping functions
- AimsCommon.js—general utility functions
- AimsXML.js—functions for basic XML communication with the servers
- AimsLayers.js—functions for managing map layers
- AimsDHTML.js—functions for creating and using Cascading Style Sheets (layers in Netscape)
- AimsClick.js—functions that respond to clicks on the map or buttons
- AimsNavigation.js—functions for interactive map navigation such as zooming and panning
- AimsLegend.js—functions for managing the graphic legend
- AimsPrint.js—functions for creating a web page layout suitable for printouts
- AimsCustom.js—templates for adding custom functionality
- AimsGeocode.js—functions to perform address and intersection matching
- AimsIdentify.js—functions to perform basic query including Identify and Hyperlink
- AimsSelect.js—functions to perform spatial selection such as selections by rectangle and shape
- AimsQuery.js—functions to perform attribute query, for example, Query, Find, and Search tools
- AimsBuffer.js—functions to perform buffering

# Using the sample HTML Viewers

Several sample implementations of the HTML Viewer have been provided with ArcIMS 3. They demonstrate a variety of functions and GUI designs. You should have a working knowledge of creating MapServices and browsing ArcIMS web sites to work with the samples.

The JavaScript function files listed on the previous page are included with all the sample HTML Viewers. There are also a few additional JavaScript files that are also included to support particular samples.

The sample applications are Basic Map, Generic Map, HyperLink, Thematic Map, Parcels, Tracker, and Java Post.

Reference the setup instructions in your \<installationdirectory>\Samples\Viewers\HTMLSample\_setup.htm file. A description and requirements for running each sample are provided below. These descriptions assume you did a ‘Typical’ installation as defined in the topic ‘General instructions for setting up samples’. If you did a ‘Custom’ installation reference this topic for complete instructions.

## Basic Map

Description:

This sample viewer demonstrates basic calls for a map image. These calls include displaying a map and zooming and panning functions. The zooming and panning are done through several different interfaces including buttons with icons (toolbar style), buttons with text (form style), and links.

To use this sample:

1. Create an Image MapService named sanfrancisco from sf.axl.
2. In the browser, type in the URL to your host website htmlviewer directory (e.g., http:\<ArcIMS host>\website\htmlviewer).
3. Click Basic Map, then click any link across the top bar. The sanfrancisco MapService is displayed.

Web site files:

In \htmlviewer\BasicMap, most of the functions are defined in MapFrame.htm and Toolbar.htm. The MapFrame file names are organized from zero to four. Each represents one of the five links across the top bar of the application, starting from the left. For example, MapFrame\_zero.htm defines the first link across the top bar. Toolbar.htm defines the buttons in the form style layout.

The Basic Map sample uses only the following four files:

- ArcIMSParam.js—the main parameter file that the viewer uses to set up its functionality
- AimsMap.js—functions that provide basic mapping functions for the viewer
- AimsCommon.js—many generic functions used by many of the other functions in the library
- AimsXML.js—functions that provide the basic XML communication with the servers

## **Generic Map**

Description:

This sample viewer presents a dropdown list of all Image MapServices running on a server and allows the user to choose one to display. It also has an Options tool that allows the user to set a variety of properties including zoom and pan factors, color for zoom box outline and map background, selection and highlight color, North arrow style, style of the layer list, and map coordinate display.

Through customization, you can create a generic data browser that lists all available MapServices running on a particular server. This type of application can be particularly effective for Intranet applications where connecting clients know the content of the MapServices already.

To implement this solution effectively, ArcIMSpParam.js should be customized to remove specific references and map extents to individual MapServices. You must also copy the JavaScript file aimsgeneric.js to your web site /javascript folder to facilitate the generic data functionality. In the MapFrame.htm file, you must also assign the generic browser to your web site by changing the variable *aimsGenericPresent* to true. So when the web site loads, no map will initially appear, and the user will then use the newly added button on the toolbar to select a new MapService for viewing.

To use this sample:

1. No specific MapService should be defined in ArcIMSpParam.js, but at least one running Image MapService is required.
2. In the browser, type in the URL to your host website htmlviewer directory (e.g., <http://<ArcIMS host>/website/htmlviewer>).
3. Click Generic Map, choose a MapService for the main map and overview map, if desired, then click Load.

Web site files:

In \htmlviewer\Generic, there is an HTML page that defines each option, with a name that is similar to the function name in the options list. For example, setHighlightColor.htm defines the Set Highlight Color function page.

The Generic Map sample uses all the files in the JavaScript function files list, plus the following:

- AimsOptions.js—functions that create the dropdown list for the options
- AimsGeneric.js—functions that create the dropdown lists for loading the requested MapServices

## **HyperLink**

Description:

This sample viewer demonstrates a hyperlink function. It shows how to turn map features into hyperlinks that display another web page. The sample presents a custom graphic look for the area surrounding the map, and includes many of the tools from BasicMap (described above).

To use this sample:

1. Create an Image MapService named sanfrancisco from sf.axl.
2. In the browser, type in the URL to your host website htmlviewer directory (e.g., `<http://<ArcIMS host>/website/htmlviewer>`).
3. Click HyperLink. Zoom in until you see the point features displayed. Open the legend to see the symbol for art galleries. Click the HyperLink tool, then click on an art gallery to link to its home page. Note: All art galleries may not have valid links.

Web site files:

In `\htmlviewer\Hyperlink`, the black custom interface is defined in `MapFrame.htm`. Look at the commented section `Map backdrop` and `Map image`.

The HyperLink sample uses all the files in the JavaScript function files list. In particular, look at `ArcIMSParam.js`—the main parameter file that the viewer uses to set up its functionality. Look for the `UseHyperLink`, `hyperLinkLayers`, and `hyperLinkFields` parameters.

## **Thematic Map**

Description:

This sample viewer shows thematic rendering of layers and generating statistics on a field. This sample has two different graphical interfaces available. The first interface, defined by `default.htm`, is a custom graphic design that requires Internet Explorer and a screen resolution of at least 1024 x 768 pixels. The second interface, defined by `standard.htm`, is similar to the standard gray HTML viewer created by ArcIMS Designer.

To use this sample:

1. Create an Image MapService named demog from `demog.axl`.
2. In the browser, type in the URL to your host website `htmlviewer` directory (e.g., `http://<ArcIMS host>/website/htmlviewer`).
3. Click Thematic Map and try out the Classify Layers and Field Statistics tools.

Web site files:

In `\htmlviewer\Thematic`, the HTML files that include Middle, Bottom, and Frame in their names contain the definitions for the pieces of the custom graphic interface.

The Thematic Map sample uses all the files in the JavaScript function files list, plus the following:

- `AimsClassRender.js`—functions that perform the classification and rendering of layers and field statistics
- `AimsClassRenderParam.js`—defines layers and fields for the rendering. In this sample, layers and fields are from data found in the provided `\Data\SanFrancisco` directory. If you experiment with other data layers, you need to update this file with the layers and fields from your data.
- `MiniDHTML.js`—defines the DHTML needed to position the graphics in the top frame of the custom graphic interface

## **Parcels**

### Description:

This sample viewer demonstrates linking layer features to data in an external database. A layer of land parcels with matching parcel information found in an Access database is provided. This sample can be used with Active Server Pages (ASP), ColdFusion, or a Visual Basic (VB) module to communicate with the external data. Source code for the VB application is provided.

### To use this sample:

1. You will need Visual Basic, ColdFusion Server, or IIS with ASP Extension, and WebLink.ocx and IMSUtil.dll from MapObjects 2.0. WebLink.ocx and IMSUtil.dll allow VB to communicate with the ArcIMS Application Server.
2. If using ColdFusion or ASP, an ODBC connection to external data is required. Create an ODBC connection as a System DSN, using the driver for Access, set the Data Source Name to “Downtown”, and set the database to downtown.mdb in the \Data\downtown directory. If using VB with an Access database, the ODBC connection is not required.
3. If using VB, make the following changes: In the \Parcels\SQLSend\sendSQL.ini file, change the ServerHostURL to match your configuration. In the \Parcels\aimsDBparam.js file, change dbDatabase to the path of the downtown.mdb file, e.g. \Data\Downtown\downtown.mdb.
4. Create an Image MapService named “parcels” from parcels.axl.
5. In the browser, type in the URL to your host website htmlviewer directory (e.g., <http://<ArcIMS host>/website/htmlviewer>). The default page is for the ColdFusion version.
6. Click Parcels, then Zoom in and try the Identify, Search by Address, and Search by Owner tools. The data returned is from the downtown.mdb database.

### Web site files:

In \htmlviewer\Parcels, the HTML files typically appear in sets, one for each of the development environments supported. ColdFusion files have a .CFM extension, ASP files have an .ASP extension, and the VB implementation uses HTML files (.HTM extension). There is also a directory named SQLSend that contains the VB application code.

The Parcels sample uses all the files in the JavaScript function files list, plus the following:

- AimsDB.js—functions that perform the linking of the layer to the external table
- AimsDBparam.js—defines fields and tables for linking

## **Tracker**

### Description:

This sample viewer demonstrates tracking the position of a moving object. This sample is dependent on a coordinate server that can send requests for object coordinates and receive coordinate responses. Source code for a coordinate server written in Visual Basic has been provided.

To use this sample:

1. Visual Basic, and WebLink.ocx and IMSUtil.dll from MapObjects 2 are required to run the Mover application. WebLink.ocx and IMSUtil.dll allow Visual Basic to communicate with the ArcIMS Application Server.
2. In \htmlviewer\Tracker\Mover\mover.ini, change the ServerHostURL to match your configuration. Also, confirm the location defined in File\_1 and File\_2, and change if necessary.
3. Start the Mover application in \htmlviewer\Tracker\Mover.
4. Create an Image MapService named parcels from parcels.axl.
5. In the browser, type in the URL to your host website htmlviewer directory (e.g., http://<ArcIMS host>\website\htmlviewer).
6. Click Tracker, then click Refresh Layer and set 10 seconds and watch the path of movement being drawn. Under Follow Object, choose one of the cars, and click Follow.

Web site files:

In \htmlviewer\Tracker, the directory named Mover includes a file named Route\_1.XY that contains a series of coordinates that feed into the VB application and define the path.

The Tracker sample uses all the files in the JavaScript function files list, plus the following:

- AimsTracker.js—sets up the protocol for sending coordinate requests and getting coordinate responses. Plots the path created by the coordinates on the map.
- AimsTrackerParam.js—defines color and size of path and object as well as other path parameters.

## **Java Post**

Description:

This sample viewer demonstrates the use of a Java applet inside the HTML Viewer. The applet communicates through ArcXML to the ArcIMS Application Server instead of the ArcIMS Servlet Connector. The applet uses Java 1.1 instead of Java 2, and therefore does not require the JRE Plug-in. The sample supports the same functions as the HTML Viewer created by ArcIMS Designer, but can be extended with Java 1.1.

To use this sample:

1. Create an Image MapService named sanfrancisco from sf.axl.
2. In the browser, type in the URL to your host website htmlviewer directory (e.g., <http://<ArcIMS host>/website/htmlviewer>).
3. Click Java Post and try out the tools on the viewer. The functions work the same as the standard HTML Viewer, but the communication is different based on a different implementation of the sendToServer function.

Web site files:

In this sample, the file \htmlviewer\JavaPost\AppletFrame.htm loads the applet instead of jsForm.htm, which is used by the standard HTML Viewer and the other samples. The MapFrame.htm contains the sendToServer function. The sendToServer function calls the Java applet to handle the communication instead of the standard posting in HTML. The directory named \java includes the Java applet and source code.

# HTML Viewer JavaScript Library

# 3

## IN THIS CHAPTER

- **Organization of the HTML Viewer JavaScript Library**
- **JavaScript functions**
- **JavaScript global variables**

The HTML Viewer provides a framework for the map, toolbar, legend, overview map, and other graphical portions of your ArcIMS web sites. You can easily customize the web sites by accessing the HTML Viewer JavaScript library of functions and global variables.

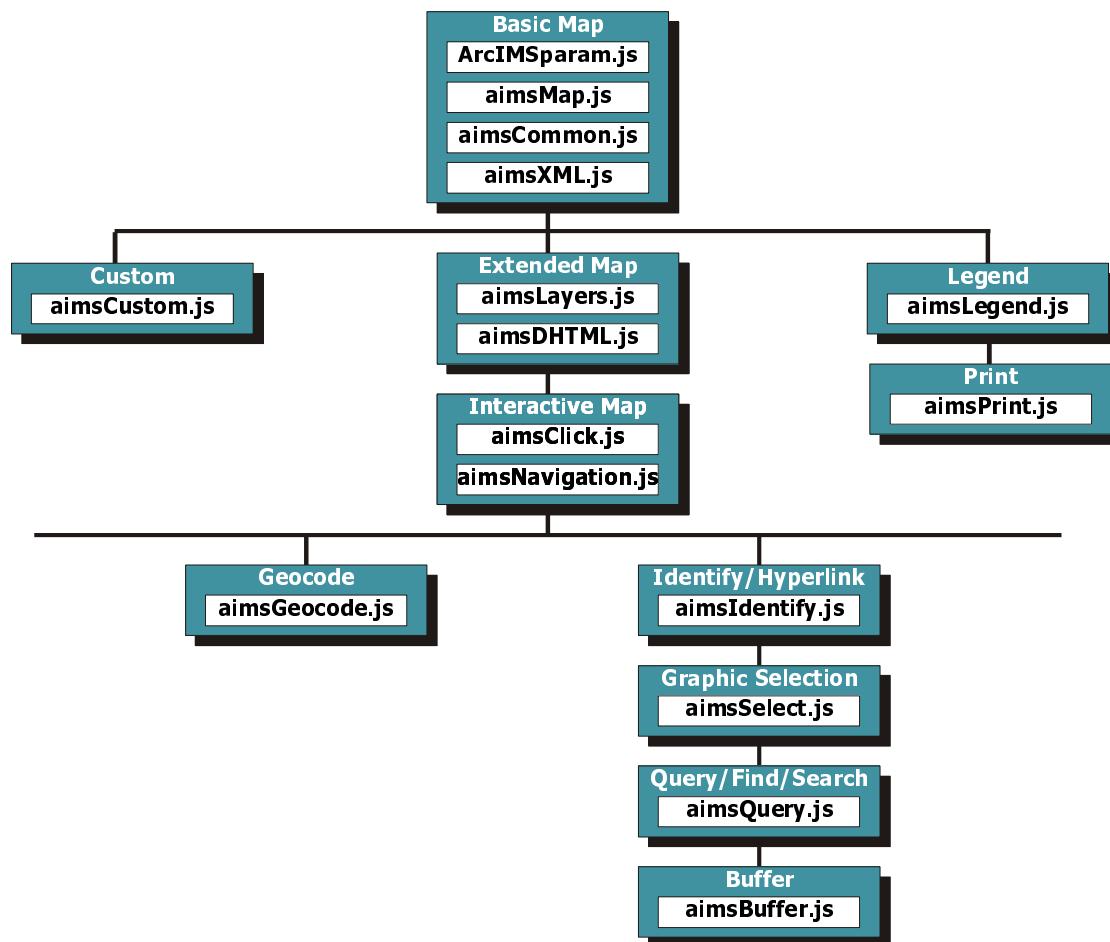
This chapter provides an overview and references to the JavaScript functions and global variables contained in the HTML Viewer.

The chapter is divided into three sections—a conceptual overview, the JavaScript functions, and the JavaScript global variables. The functions and global variables sections include charts arranged alphabetically by name and by category. The main component of these sections are the descriptions (listed alphabetically by name) which contain the specific information needed for customization.

This chapter assumes a familiarity with HTML and JavaScript.

# Organization of the HTML Viewer JavaScript Library

The HTML Viewer uses a library of JavaScript functions found in several files located in the javascript subdirectory. These files are grouped into categories based on viewer functionality—Basic Map, Custom, Extended Map, Interactive Map, Legend, Print, Geocode, IDentify/Hyperlink, Graphic Selection, Query/Find/Search, and Buffer. The categories, files, and hierarchy of dependency are illustrated below.



## **Basic Map**

The Basic Map category consists of four files: ArcIMSPparam.js, aimMap.js, aimsCommon.js, and aimsXML.js. These files are required by the viewer and provide the minimum functionality necessary for displaying map images.

ArcIMSPparam.js is primarily a parameter file that the viewer uses to set up its configuration.

AimsMap.js contains functions that provide basic mapping capabilities for the viewer.

AimsCommon.js contains common utilities used by the various functions in the library.

AimsXML.js contains functions that provide the basic XML communication with the servers.

## **Custom**

The Custom category consists of a single file: aimsCustom.js. Use of this file requires the files from the Basic Map category.

AimsCustom.js contains templates for adding custom functions to the viewer.

## **Extended Map**

The Extended Map category consists of two files: aimsLayers.js and aimsDHTML.js. These require the Basic Map files and extend the capabilities of the viewer to provide the foundation required before functions for user interactivity can be added.

AimsLayers.js contains functions for managing map layers.

AimsDHTML.js contains functions for creating and using style sheets (layers in Netscape).

## **Interactive Map**

The Interactive Map category consists of two files: aimsClick.js and aimsNavigation.js. These add user interactivity to the viewer and require the files found in the Basic Map and Extended Map categories.

AimsClick.js contains functions that respond to clicks on the map or the tool buttons.

AimsNavigation.js contains functions for map navigation such as zooming and panning.

## **Legend**

The Legend category consists of a single file: aimsLegend.js. Use of this file requires the files from the Basic Map category.

AimsLegend.js contains functions that are associated with the graphic legend.

## **Print**

The Print category consists of a single file: aimsPrint.js. Use of this file requires the files from the Basic Map category.

AimsPrint.js contains functions that are associated with creating a web page layout for printouts.

## **Geocode**

The Geocode category consists of a single file: aimsGeocode.js. Use of this file requires the files from the Basic Map, Extended Map, and Interactive Map categories.

AimsGeocode.js contains functions to allow address and intersection matching capabilities to the viewer.

## **IDeIdentify/Hyperlink**

The IDIdentify/Hyperlink category consists of a single file: aimsIDIdentify.js. Use of this file requires the files from the Basic Map, Extended Map, and Interactive Map categories.

AimsIDIdentify.js contains functions to add basic query capabilities to the viewer. IDIdentify and Hyperlink are created by functions in this file.

## **Graphic Selection**

The Graphic Selection category consists of a single file: aimsSelect.js. Use of this file requires the files from the Basic Map, Extended Map, Interactive Map, and IDIdentify/Hyperlink categories.

AimsSelect.js contains functions to add spatial selection capabilities to the viewer. Selections by shape (rectangle, line, or polygon) are created by functions in this file.

## **Query/Find/Search**

The Query/Find/Search category consists of a single file: aimsQuery.js. Use of this file requires the files from the Basic Map, Extended Map, Interactive Map, IDIdentify/Hyperlink, and Select categories.

AimsQuery.js contains functions to add attribute query capabilities to the viewer. Queries defined by the Query, Find, and Search tools are created by functions in this file.

## **Buffer**

The Buffer category consists of a single file: aimsBuffer.js. Use of this file requires the files from the Basic Map, Extended Map, Interactive Map, IDIdentify/Hyperlink, Select, and Query categories.

AimsBuffer.js contains functions to add buffering capabilities to the viewer.

## JavaScript functions by name (add-full)

Function Name	File	Category
addBufferToMap	aimsBuffer.js	Buffer
addCustomToMap1	aimsCustom.js	Custom
addCustomToMap2	aimsCustom.js	Custom
addCustomToMap3	aimsCustom.js	Custom
addCustomToMap4	aimsCustom.js	Custom
addLegendToMap	aimsLegend.js	Legend
addSelectToMap	aimsSelect.js	Select
afterMapRefresh	aimsMap.js	Basic Map
beforeMapRefresh	aimsMap.js	Basic Map
boxIt	aimsDHTML.js	Extended Map
bufferIt	aimsBuffer.js	Buffer
calcDistance	aimsMap.js	Basic Map
checkCurrentExtent	aimsCommon.js	Basic Map
checkFullExtent	aimsMap.js	Basic Map
checkHyperLinkLayer	aimsIDIdentify.js	IDIdentify/Hyperlink
checkParams	ArcIMSpParam.js	Basic Map
checkSelected	aimsIDIdentify.js	IDIdentify/Hyperlink
checkStoredQueries	aimsQuery.js	Query
chkMouseUp	aimsClick.js	Interactive Map
clearError	aimsCommon.js	Basic Map
clearLeadingSpace	aimsCommon.js	Basic Map
clearSelection	aimsSelect.js	Select
clickAddPoint	aimsClick.js	Interactive Map
clickFunction	aimsClick.js	Interactive Map
clipLayer	aimsDHTML.js	Extended Map
convertHexToDec	aimsCommon.js	Basic Map
convertUnits	aimsMap.js	Basic Map
createLayer	aimsDHTML.js	Extended Map
customMapTool	aimsCustom.js	Custom
deleteClick	aimsClick.js	Interactive Map
displayAttributeData	aimsIDIdentify.js	IDIdentify/Hyperlink
extractIt	aimsCustom.js	Custom
findForm	aimsQuery.js	Query
fixSingleQuotes	aimsCommon.js	Basic Map
fullExtent	aimsMap.js	Basic Map

## JavaScript functions by name (getA-getXML)

Function Name	File	Category
getAllFieldValues	aimsCommon.js	Basic Map
getBufferAttributeData	aimsBuffer.js	Buffer
getCommandLineParams	aimsMap.js	Basic Map
getEnvelopeXYs	aimsXML.js	Basic Map
getFieldNames	aimsCommon.js	Basic Map
getFieldValues	aimsCommon.js	Basic Map
getFind	aimsQuery.js	Query
getGeocodeLayers	aimsGeocode.js	Geocode
getGeocodeParams	aimsGeocode.js	Geocode
getIDValue	aimsCommon.js	Basic Map
getImageXY	aimsNavigation.js	Interactive Map
getInsideString	aimsCommon.js	Basic Map
getLayer	aimsDHTML.js	Extended Map
getLayerFieldNames	aimsLayers.js	Extended Map
getLayerFieldPrecisions	aimsLayers.js	Extended Map
getLayerFieldSizes	aimsLayers.js	Extended Map
getLayerFieldTypes	aimsLayers.js	Extended Map
getLayers	aimsLayers.js	Extended Map
getLegend	aimsLegend.js	Legend
getLegendURL	aimsXML.js	Basic Map
getMapHeight	aimsCommon.js	Basic Map
getMapWidth	aimsCommon.js	Basic Map
getMapXY	aimsNavigation.js	Interactive Map
getMoreData	aimsSelect.js	Select
getMouse	aimsNavigation.js	Interactive Map
getOVImageXY	aimsNavigation.js	Interactive Map
getOVXYs	aimsXML.js	Basic Map
getPath	aimsMap.js	Basic Map
getPrintLegend	aimsPrint.js	Print
getPrintMap	aimsPrint.js	Print
getPrintOV	aimsPrint.js	Print
getScaleBarDistance	aimsMap.js	Basic Map
getStartExtent	aimsCommon.js	Basic Map
getStoredQueries	aimsQuery.js	Query
getXMLErrorMessage	aimsXML.js	Basic Map

## JavaScript functions by name (getURL-put)

Function Name	File	Category
getURL	aimsXML.js	Basic Map
getXYs	aimsXML.js	Basic Map
hasLayer	aimsMap.js	Basic Map
hideLayer	aimsDHTML.js	Extended
hideRetrieveData	aimsMap.js	Basic Map
hideRetrieveMap	aimsMap.js	Basic Map
hyperLink	aimsIDeIdentify.js	IDeIdentify/Hyperlink
IDeIdentify	aimsIDeIdentify.js	IDeIdentify/Hyperlink
isVisible	aimsDHTML.js	Extended Map
justGetFeatureCount	aimsCommon.js	Basic Map
justGetFieldValue	aimsCommon.js	Basic Map
justGetMap	aimsXML.js	Basic Map
justGetValue	aimsCommon.js	Basic Map
makeXMLsafe	aimsCommon.js	Basic Map
mapTool	aimsClick.js	Interactive Map
moveLayer	aimsDHTML.js	Extended Map
numberorder	aimsCommon.js	Basic Map
ovMap2Click	aimsNavigation.js	Interactive Map
ovMapClick	aimsNavigation.js	Interactive Map
pan	aimsNavigation.js	Interactive Map
panButton	aimsMap.js	Basic Map
panMouse	aimsNavigation.js	Interactive Map
parseEntity	aimsCommon.js	Basic Map
parseFieldSamples	aimsQuery.js	Query
parseGeocodeLayers	aimsGeocode.js	Geocode
parseGeocodeParams	aimsGeocode.js	Geocode
parseGeocodeResults	aimsGeocode.js	Geocode
parseHyperLink	aimsIDeIdentify.js	IDeIdentify/Hyperlink
parseRecordString	aimsCommon.js	Basic Map
parseStoredQueries	aimsQuery.js	Query
printIt	aimsPrint.js	Print
processStartExtent	aimsCommon.js	Basic Map
processXML	aimsXML.js	Basic Map
putExtentOnOVMap	aimsDHTML.js	Extended Map

## JavaScript functions by name (query-start)

Function Name	File	Category
queryForm	aimsQuery.js	Query
reloadApp	aimsCommon.js	Basic Map
replaceLayerContent	aimsDHTML.js	Extended Map
replacePlus	aimsCommon.js	Basic Map
resetClick	aimsClick.js	Interactive Map
resetError	aimsCommon.js	Basic Map
saveLastExtent	aimsMap.js	Basic Map
select	aimsSelect.js	Select
sendCustomToServer	aimsXML.js	Basic Map
sendMapXML	aimsXML.js	Basic Map
sendQueryString	aimsQuery.js	Query
sendShapeSelect	aimsSelect.js	Select
sendStoredQuery	aimsQuery.js	Query
sendToServer	aimsXML.js	Basic Map
setActiveLayer	aimsLayers.js	Extended Map
setClip	aimsNavigation.js	Interactive Map
setExtent	aimsMap.js	Basic Map
setFullExtent	aimsMap.js	Basic Map
setLayerBackgroundColor	aimsDHTML.js	Extended Map
setLayerFields	aimsLayers.js	Extended Map
setupGeocode	aimsGeocode.js	Geocode
setZoomColor	aimsNavigation.js	Interactive Map
showHighlight	aimsSelect.js	Select
showLayer	aimsDHTML.js	Extended Map
showLayerInfo	aimsLayers.js	Extended Map
showLegend	aimsLegend.js	Legend
showRetrieveData	aimsMap.js	Basic Map
showRetrieveMap	aimsMap.js	Basic Map
startExtent	aimsMap.js	Basic Map
startMap	aimsCommon.js	Basic Map
startPan	aimsNavigation.js	Interactive Map
startSelectBox	aimsSelect.js	Select
startUp	aimsCommon.js	Basic Map
startZoomBox	aimsNavigation.js	Interactive Map
startZoomOutBox	aimsNavigation.js	Interactive Map

## JavaScript functions by name (stop-writeShape)

Function Name	File	Category
stopPan	aimsNavigation.js	Interactive Map
stopSelectBox	aimsSelect.js	Select
stopZoomBox	aimsNavigation.js	Interactive Map
stopZoomOutBox	aimsNavigation.js	Interactive Map
storedQueryForm	aimsQuery.js	Query
swapQuotes	aimsCommon.js	Basic Map
swapStuff	aimsCommon.js	Basic Map
tempGetSamples	aimsQuery.js	Query
toggleOVMap	aimsDHTML.js	Extended Map
untag	aimsCommon.js	Basic Map
updateMeasureBox	aimsClick.js	Interactive Map
useCustomFunction	aimsCustom.js	Custom
writeBlankMapXML	aimsXML.js	Basic Map
writeBufferForm	aimsBuffer.js	Buffer
writeEnvelopeBufferXML	aimsBuffer.js	Buffer
writeEnvelopeXML	aimsSelect.js	Select
writeFieldSample	aimsQuery.js	Query
writeFindRequest	aimsQuery.js	Query
writeGeocodeXML	aimsGeocode.js	Geocode
writeGetBufferData	aimsBuffer.js	Buffer
writeGetFeatures	aimsIDeIdentify.js	IDeIdentify/Hyperlink
writeGetFeatures2	aimsSelect.js	Select
writeGetFeatures3	aimsSelect.js	Select
writelIdentifyXML	aimsIDeIdentify.js	IDeIdentify/Hyperlink
writeLayerListForm	aimsLayers.js	Extended Map
writeModeFrame	aimsCommon.js	Basic Map
writeModeLayers	aimsCommon.js	Basic Map
writeOverlayXML	aimsClick.js	Interactive Map
writeOVXML	aimsXML.js	Basic Map
writePrintPage	aimsPrint.js	Print
writeQueryBufferXML	aimsBuffer.js	Buffer
writeQueryForm	aimsQuery.js	Query
writeQueryXML	aimsQuery.js	Query
writeShapeBufferXML	aimsBuffer.js	Buffer
writeShapeSelect	aimsSelect.js	Select

## JavaScript functions by name (writeStored-zoom)

Function Name	File	Category
writeStoredQueryForm	aimsQuery.js	Query
writeXML	aimsXML.js	Basic Map
zoomBack	aimsMap.js	Basic Map
zoomButton	aimsMap.js	Basic Map
zoomin	aimsNavigation.js	Interactive Map
zoomout	aimsNavigation.js	Interactive Map
zoomScale	aimsMap.js	Basic Map
zoomToEnvelope	aimsMap.js	Basic Map
zoomToPoint	aimsMap.js	Basic Map

## JavaScript functions by Category (Basic Map)

Category	File	Function Name
Basic Map	aimsMap.js	saveLastExtent
Basic Map	aimsMap.js	fullExtent
Basic Map	aimsMap.js	startExtent
Basic Map	aimsMap.js	zoomBack
Basic Map	aimsMap.js	zoomToPoint
Basic Map	aimsMap.js	zoomToEnvelope
Basic Map	aimsMap.js	zoomScale
Basic Map	aimsMap.js	getCommandLineParams
Basic Map	aimsMap.js	getPath
Basic Map	aimsMap.js	hasLayers
Basic Map	aimsMap.js	showRetrieveData
Basic Map	aimsMap.js	hideRetrieveData
Basic Map	aimsMap.js	showRetrieveMap
Basic Map	aimsMap.js	hideRetrieveMap
Basic Map	aimsMap.js	getScaleBarDistance
Basic Map	aimsMap.js	calcDistance
Basic Map	aimsMap.js	convertUnits
Basic Map	aimsMap.js	setExtent
Basic Map	aimsMap.js	setFullExtent
Basic Map	aimsMap.js	beforeMapRefresh
Basic Map	aimsMap.js	afterMapRefresh
Basic Map	aimsMap.js	zoomButton
Basic Map	aimsMap.js	panButton
Basic Map	aimsMap.js	checkFullExtent
Basic Map	aimsXML.js	sendToServer
Basic Map	aimsXML.js	sendCustomToServer
Basic Map	aimsXML.js	sendMapXML
Basic Map	aimsXML.js	processXML
Basic Map	aimsXML.js	writeBlankMapXML
Basic Map	aimsXML.js	writeXML
Basic Map	aimsXML.js	writeOVXML
Basic Map	aimsXML.js	getXYs
Basic Map	aimsXML.js	getOVXYs
Basic Map	aimsXML.js	getURL
Basic Map	aimsXML.js	getLegendURL

## JavaScript functions by Category (Basic Map continued)

<b>Category</b>	<b>File</b>	<b>Function Name</b>
Basic Map	aimsXML.js	justGetMapextent
Basic Map	aimsXML.js	getEnvelopeXYs
Basic Map	aimsXML.js	getXMLErrorMessage
Basic Map	ArcIMSparam.js	checkParams
Basic Map	aimsCommon.js	getInsideString
Basic Map	aimsCommon.js	justGetValue
Basic Map	aimsCommon.js	getIDValue
Basic Map	aimsCommon.js	getFieldValues
Basic Map	aimsCommon.js	getFieldNames
Basic Map	aimsCommon.js	parseRecordString
Basic Map	aimsCommon.js	fixSingleQuotes
Basic Map	aimsCommon.js	untag
Basic Map	aimsCommon.js	clearLeadingSpace
Basic Map	aimsCommon.js	reloadApp
Basic Map	aimsCommon.js	resetError
Basic Map	aimsCommon.js	clearError
Basic Map	aimsCommon.js	swapStuff
Basic Map	aimsCommon.js	convertHexToDec
Basic Map	aimsCommon.js	swapQuotes
Basic Map	aimsCommon.js	writeModeLayers
Basic Map	aimsCommon.js	writeModeFrame
Basic Map	aimsCommon.js	checkCurrentExtent
Basic Map	aimsCommon.js	getMapHeight
Basic Map	aimsCommon.js	getMapWidth
Basic Map	aimsCommon.js	startMap
Basic Map	aimsCommon.js	processStartExtent
Basic Map	aimsCommon.js	getStartExtent
Basic Map	aimsCommon.js	startUp
Basic Map	aimsCommon.js	justGetFieldValue
Basic Map	aimsCommon.js	justGetFeatureCount
Basic Map	aimsCommon.js	getAllFieldValues
Basic Map	aimsCommon.js	numberorder
Basic Map	aimsCommon.js	parseEntity
Basic Map	aimsCommon.js	makeXMLsafe
Basic Map	aimsCommon.js	replacePlus

## JavaScript functions by Category (Buffer-Extended Map)

Category	File	Function Name
Buffer	aimsBuffer.js	writeBufferForm
Buffer	aimsBuffer.js	bufferIt
Buffer	aimsBuffer.js	addBufferToMap
Buffer	aimsBuffer.js	writeGetBufferData
Buffer	aimsBuffer.js	writeQueryBufferXML
Buffer	aimsBuffer.js	writeShapeBufferXML
Buffer	aimsBuffer.js	writeEnvelopeBufferXML
Buffer	aimsBuffer.js	getBufferAttributeData
Custom	aimsCustom.js	customMapTool
Custom	aimsCustom.js	useCustomFunction
Custom	aimsCustom.js	addCustomToMap1
Custom	aimsCustom.js	addCustomToMap2
Custom	aimsCustom.js	addCustomToMap3
Custom	aimsCustom.js	addCustomToMap4
Custom	aimsCustom.js	extractIt
Extended Map	aimsLayers.js	setActiveLayer
Extended Map	aimsLayers.js	writeLayerListForm
Extended Map	aimsLayers.js	getLayers
Extended Map	aimsLayers.js	setLayerFields
Extended Map	aimsLayers.js	getLayerFieldNames
Extended Map	aimsLayers.js	getLayerFieldTypes
Extended Map	aimsLayers.js	getLayerFieldSizes
Extended Map	aimsLayers.js	getLayerFieldPrecisions
Extended Map	aimsLayers.js	showLayerInfo
Extended Map	aimsDHTML.js	createLayer
Extended Map	aimsDHTML.js	getLayer
Extended Map	aimsDHTML.js	isVisible
Extended Map	aimsDHTML.js	moveLayer
Extended Map	aimsDHTML.js	setLayerBackgroundColor
Extended Map	aimsDHTML.js	hideLayer
Extended Map	aimsDHTML.js	showLayer
Extended Map	aimsDHTML.js	clipLayer
Extended Map	aimsDHTML.js	replaceLayerContent
Extended Map	aimsDHTML.js	toggleOVMMap
Extended Map	aimsDHTML.js	putExtentOnOVMMap
Extended Map	aimsDHTML.js	boxIt

## JavaScript functions by Category (Geocode-IDentify)

Category	File	Function Name
Geocode	aimsGeocode.js	setupGeocode
Geocode	aimsGeocode.js	getGeocodeLayers
Geocode	aimsGeocode.js	getGeocodeParams
Geocode	aimsGeocode.js	writeGeocodeXML
Geocode	aimsGeocode.js	parseGeocodeLayers
Geocode	aimsGeocode.js	parseGeocodeParams
IDeIdentify/Hyperlink	aimsIDeIdentify.js	identify
IDeIdentify/Hyperlink	aimsIDeIdentify.js	hyperLink
IDeIdentify/Hyperlink	aimsIDeIdentify.js	writeGetFeatures
IDeIdentify/Hyperlink	aimsIDeIdentify.js	writeIDeIdentifyXML
IDeIdentify/Hyperlink	aimsIDeIdentify.js	displayAttributeData
IDeIdentify/Hyperlink	aimsIDeIdentify.js	parseHyperLink
IDeIdentify/Hyperlink	aimsIDeIdentify.js	checkSelected
IDeIdentify/Hyperlink	aimsIDeIdentify.js	checkHyperLinkLayer

## JavaScript functions by Category (Interactive Map)

Category	File	Function Name
Interactive Map	aimsClick.js	clickAddPoint
Interactive Map	aimsClick.js	resetClick
Interactive Map	aimsClick.js	deleteClick
Interactive Map	aimsClick.js	clickFunction
Interactive Map	aimsClick.js	chkMouseUp
Interactive Map	aimsClick.js	mapTool
Interactive Map	aimsClick.js	updateMeasureBox
Interactive Map	aimsClick.js	writeOverlayXML
Interactive Map	aimsNavigation.js	getMapXY
Interactive Map	aimsNavigation.js	getImageXY
Interactive Map	aimsNavigation.js	getOVIImageXY
Interactive Map	aimsNavigation.js	ovMapClick
Interactive Map	aimsNavigation.js	ovMap2Click
Interactive Map	aimsNavigation.js	zoomin
Interactive Map	aimsNavigation.js	zoomout
Interactive Map	aimsNavigation.js	getMouse
Interactive Map	aimsNavigation.js	startZoomBox
Interactive Map	aimsNavigation.js	stopZoomBox
Interactive Map	aimsNavigation.js	startZoomOutBox
Interactive Map	aimsNavigation.js	stopZoomOutBox
Interactive Map	aimsNavigation.js	setClip
Interactive Map	aimsNavigation.js	startPan
Interactive Map	aimsNavigation.js	stopPan
Interactive Map	aimsNavigation.js	panMouse
Interactive Map	aimsNavigation.js	pan
Interactive Map	aimsNavigation.js	setZoomColor

## JavaScript functions by Category (Legend-Select)

<b>Category</b>	<b>File</b>	<b>Function Name</b>
Legend	aimsLegend.js	getLegend
Legend	aimsLegend.js	showLegend
Legend	aimsLegend.js	addLegendToMap
Print	aimsPrint.js	printIt
Print	aimsPrint.js	getPrintMap
Print	aimsPrint.js	getPrintOV
Print	aimsPrint.js	getPrintLegend
Print	aimsPrint.js	writePrintPage
Query	aimsQuery.js	queryForm
Query	aimsQuery.js	findForm
Query	aimsQuery.js	sendQueryString
Query	aimsQuery.js	writeQueryXML
Query	aimsQuery.js	writeFieldSample
Query	aimsQuery.js	getFind
Query	aimsQuery.js	writeFindRequest
Query	aimsQuery.js	parseFieldSamples
Query	aimsQuery.js	writeQueryForm
Query	aimsQuery.js	tempGetSamples
Query	aimsQuery.js	getStoredQueries
Query	aimsQuery.js	parseStoredQueries
Query	aimsQuery.js	storedQueryForm
Query	aimsQuery.js	writeStoredQueryForm
Query	aimsQuery.js	sendStoredQuery
Query	aimsQuery.js	checkStoredQueries
Select	aimsSelect.js	select
Select	aimsSelect.js	startSelectBox
Select	aimsSelect.js	stopSelectBox
Select	aimsSelect.js	writeGetFeatures2
Select	aimsSelect.js	writeGetFeatures3
Select	aimsSelect.js	writeEnvelopeXML
Select	aimsSelect.js	writeShapeSelect
Select	aimsSelect.js	sendShapeSelect
Select	aimsSelect.js	getMoreData
Select	aimsSelect.js	showHighlight
Select	aimsSelect.js	clearSelection
Select	aimsSelect.js	addSelectToMap

## **addBufferToMap**

### **Description:**

Adds buffering instructions to ArcXML map image request.

Uses: bufferdistance, selectionMode, drawTargetLayer, bufferSmoothEdges, ScaleBarUnits, setQueryString, limitRight, limitBottom, limitLeft, limitTop, bufferTargetLayer, bufferTargetLayerIndex, useLimitExtent, clickType, clickCount, and selectEnvelope.

Called by: writeXML function when aimsBuffer.js has been loaded and showBuffer is true.

### **Category:**

Buffer

### **File:**

aimsBuffer.js

### **Syntax:**

addBufferToMap()

### **Arguments:**

None

### **Returned Value:**

String	Returns string with buffering instructions to be inserted into ArcXML map image request.
--------	--

### **See Also**

[writeXML](#)

## **addCustomToMap1, addCustomToMap2, addCustomToMap3, addCustomToMap4**

### **Description:**

Adds custom instructions to ArcXML map image request. By default these functions do not contain any code that creates any custom instructions and are provided for customizing the image request. They can be modified to add desired functions to the MapXML request at specific points. addCustomToMap1() occurs between selection and geocode. AddCustomToMap2() occurs between clickpoints and copyright. AddCustomToMap3() occurs under modeOnMap. AddCustomToMap4() occurs on top of everything.

Called by: writeXML function if aimsCustom.js has been loaded.

### **Category:**

Custom

### **File:**

aimsCustom.js

### **Syntax:**

addCustomToMap1(), addCustomToMap2(), addCustomToMap3(), addCustomToMap4()

### **Arguments:**

None

### **Returned Value:**

String Returns string with custom instructions to be inserted into ArcXML map image request.  
By default an empty string is returned.

### **See Also**

[writeXML](#)

## **addLegendToMap**

### **Description:**

Adds legend creation instructions to ArcXML map image request.

Uses: legHeight, legTitle, legFont, legWidth, and drawLegendOnly.

Called by: writeXML function if aimsLegend.js has been loaded and legendVisible is true.

### **Category:**

Legend

### **File:**

aimsLegend.js

### **Syntax:**

addLegendToMap()

### **Arguments:**

None

### **Returned Value:**

String      Returns string with legend image generation instructions to be inserted into ArcXML map image request.

### **See Also**

[writeXML](#)

## **addSelectToMap**

### **Description:**

Adds selection instructions to ArcXML map image request.

Uses: selectCount, showselectedFeatures, selectionMode, setQueryString, useLimitExtent, limitRight, limitLeft, limitBottom, limitTop, clickCount, transparentLevel, clickType, selectType, ActiveLayerIndex, selectEnvelope, selectBlurb, ActiveLayer, selectColor, highlightedOne, highlightColor.

Uses: clickPointX, clickY, and LayerNamearrays.

Called by: writeXML function if aimsSelect.js has been l.

### **Category:**

Select

### **File:**

aimsSelect.js

### **Syntax:**

addSelectToMap()

### **Arguments:**

None

### **Returned Value:**

String      Returns string with selection instructions to be inserted into ArcXML map image request. If no features are selected, an empty string is returned.

### **See Also**

[writeXML](#)

## **afterMapRefresh**

### **Description:**

Custom instructions to be executed after a request for a new map image. It is suggested that a duplicate definition be put into MapFrame.htm after the line loading aimsMap.js if modification of aimsMap.js is not desired.

Called by: processXML function.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

afterMapRefresh()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

processXML

## **beforeMapRefresh**

### **Description:**

Custom instructions to be executed before a request for a new map image. It is suggested that a duplicate definition be put into MapFrame.htm after the line loading aimsMap.js if modification of aimsMap.js is not desired.

Called by :sendMapXML function.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

beforeMapRefresh()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

[sendMapXML](#)

## **boxIt**

### **Description:**

Creates the Zoom/Select box on the map image. The box is created by four cascading style sheets (layers in Netscape) overlaying the map image.

Calls: moveLayer or clipLayer in aimsDHTML.js.

Called by: startZoomBox, startZoomOutBox, setClip, and startSelectBox functions.

### **Category:**

Extended Map

### **File:**

aimsDHTML.js

### **Syntax:**

boxIt(theLeft,theTop,theRight,theBottom)

### **Arguments:**

theLeft	Numeric representing pixel x-coordinate of left edge of box.
theTop	Numeric representing pixel y-coordinate of top edge of box.
theRight	Numeric representing pixel x-coordinate of right edge of box.
theBottom	Numeric representing pixel y-coordinate of bottom edge of box.

### **Returned Value:**

None

### **See Also**

startZoomBox	startZoomOutBox
setClip	startSelectBox
moveLayer	clipLayer

## **bufferIt**

### **Description:**

Sets showBuffer to true so that buffer instructions will be added to the map image request.

Calls: sendMapXML in aimsXML.js and hidelay in aimsDHTML.js.

Called by: form created by writeBufferForm.

### **Category:**

Buffer

### **File:**

aimsBuffer.js

### **Syntax:**

bufferIt()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

[writeBufferForm](#)

[hidelay](#)

## **calcDistance**

### **Description:**

Calculates distance from last user click on map to specified position.

Uses: clickPointX and clickPointY arrays, clickCount, MapUnits, and ScaleBarUnits.

Calls: updateMeasureBox and convertUnits.

Called by: getMouse for Measure mode.

### **Category:**

Basic Map (called by Interactive Map functions)

### **File:**

aimsMap.js

### **Syntax:**

calcDistance(mX,mY)

### **Arguments:**

mX            Numeric representing map x-coordinate in map units.

mY            Numeric representing map y-coordinate in map units.

### **Returned Value:**

None

### **See Also**

getMouse            convertunits

updateMeasureBox

## **checkCurrentExtent**

### **Description:**

Displays current extent coordinates in an alert box.

Uses: left, top, right, bottom, xDistance, and fullWidth.

Used for: debugging.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

checkCurrentExtent()

### **Arguments:**

None

### **Returned Value:**

None

## **checkFullExtent**

### **Description:**

Checks new extent for coordinates beyond extent limit and modifies any coordinates outside of extent limit to limit edge if necessary.

Uses: left, top, right, bottom, xDistance, yDistance, fullWidth, fullHeight, enforceFullExtent, imageLimitLeft, imageLimitRight, imageLimitTop, and imageLimitBottom.

Called by: zoomButton and panButton in aimsMap.js, and by stopPan in aimsNavigation.js.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

checkFullExtent()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

zoomButton    stopPan

panButton

## **checkHyperLinkLayer**

### **Description:**

Checks if a layer is configured for hyperlinking and sets it as the current hyperlink layer. Updates currentHyperLinkLayer and currentHyperLinkField.

Uses: hyperLinkLayers and hyperLinkFields arrays.

Called by: clickFunction in aimsClick.js.

### **Category:**

IDeIdentify/Hyperlink

### **File:**

aimsIDeIdentify.js

### **Syntax:**

checkHyperLinkLayer(layerIndex)

### **Arguments:**

layerIndex      Number representing the index of the layer. The topmost layer is zero.

### **Returned Value:**

Boolean      True or False

### **See Also**

clickFunction

## **checkParams**

### **Description:**

Checks various parameters on startup.

Calls: getPath, getCommonLineParams in aimsMap.js, clickFunction in aimsClick.js, and startMap in aimsCommon.js.

Called by: onload parameter in Frame setup in viewer.htm.

### **Category:**

Basic Map

### **File:**

ArcIMSParam.js

### **Syntax:**

checkParams()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

getPath	clickFunction
startMap	getCommonLineParams

## **checkSelected**

### **Description:**

Checks various parameters on startup.

Calls: getPath, getCommonLineParams in aimsMap.js, clickFunction in aimsClick.js, and startMap in aimsCommon.js.

Called by: onload parameter in Frame setup in viewer.htm.

### **Category:**

IDeIdentify/Hyperlink

### **File:**

aimsIDeIdentify.js

### **Syntax:**

checkSelected()

### **Arguments:**

None

### **Returned Value:**

Boolean      True or False

### **See Also**

[clickFunction](#)

## **checkStoredQueries**

### **Description:**

Checks if there are any StoredQueries in the MapService.

Sets: useStoredQuery to False if there are none.

Called by: processStartExtent in aimsCommon.js if useStoredQuery initially is True.

### **Category:**

Query

### **File:**

aimsQuery.js

### **Syntax:**

checkStoredQueries(theReply)

### **Arguments:**

theReply      String containing returned ArcXML response of service information.

### **Returned Value:**

None

### **See Also**

[processStartExtent](#)

## **chkMouseUp**

### **Description:**

Checks if the current cursor mode is ZoomIn, ZoomOut, Pan, or SelectRectangle when the mouse button is pressed and the cursor moves outside the main map Display area.

Uses: toolMode, zooming, panning, and selectBox.

Calls: stopZoomBox, stopZoomOutBox, stopPan, or stopSelectBox.

Called by: getMouse in aimsNavigation.js if the cursor moves outside the main map display when the mouse button is down.

### **Category:**

Interactive Map

### **File:**

aimsClick.js

### **Syntax:**

chkMouseUp(e)

### **Arguments:**

e              Event passed by browser.

### **Returned Value:**

Boolean       Returns False

### **See Also**

getMouse

## **clearError**

### **Description:**

Disables JavaScript error checking.

Uses: resetError to reset error checking to default.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

clearError()

### **Arguments:**

None

### **Returned Value:**

Boolean      Returns True.

### **See Also**

[resetError](#)

## **clearLeadingSpace**

### **Description:**

Removes leading spaces in field values returned in ArcXML response string.

Called by: `getBufferAttributeData` in `aimsBuffer.js`, `displayAttributeData` in `aimsIDebug.js`, and `parseFieldSamples` in `aimsQuery.js`.

### **Category:**

Basic Map

### **File:**

`aimsCommon.js`

### **Syntax:**

`clearLeadingSpace(inText)`

### **Arguments:**

`inText`      String containing ArcXML response string of field values returned from query/  
selection.

### **Returned Value:**

`String`    String containing processed text.

### **See Also**

`getBufferAttributeData`      `displayAttributeData`  
`parseFieldSamples`

## **clearSelection**

### **Description:**

Sets selection count to zero.

Uses: useTextFrame and toolMode.

Sets: selectCount to zero. Resets the selectPoints, selectLeft, selectRight, selectTop, and selectBottom arrays. Sets showBuffer to False. Sets highlightedOne to empty string. Sets legendVisible to False.

Calls: showLayer, sendToServer, and writeXML. Also calls updateMeasureBox if toolMode = 20 (Measure Mode).

Called by: clickFunction in aimsClick.js.

### **Category:**

Select

### **File:**

aimsSelect.js

### **Syntax:**

clearSelection()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

clickFunction

showLayer

updateMeasureBox

sendToServer

writeXML

# **clickAddPoint**

## **Description:**

Adds a new click location to collection of click locations.

Sets: selectCount to zero. Updates the clickPointX, clickPointY, and clickMeasure arrays.

Updates clickCount and totalMeasure.

Uses: mapX, mapY, noOverLayer, and legendVisible.

Calls: getMapXY, sendToServer, writeOverlayXML and writeXML.

Called by: clickFunction in aimsClick.js.

## **Category:**

Interactive Map

## **File:**

aimsClick.js

## **Syntax:**

clickAddPoint()

## **Arguments:**

None

## **Returned Value:**

None

## **See Also**

clickFunction    getMapXY

sendToServer    writeOverlayXML

writeXML

## **clickFunction**

### **Description:**

Sets current cursor mode.

Uses: isIE, useTextFrame, canQuery, clickType, useModeFrame, drawFloating Mode, modeLayersOn, modeRefreshMap, drawModeOnMap, useBuffer, and hasTOC.

Calls: function associated with toolName, hasLayer, sendMapXML, writeModeFrame, and writeModeLayers.

Sets: toolMode and modeBlurb. Sets globals associated with toolName.

### **Category:**

Interactive Map

### **File:**

aimsClick.js

### **Syntax:**

`clickFunction(toolName)`

### **Arguments:**

`toolName`      String containing name of cursor mode or mode that user has selected.

### **Returned Value:**

None

### **See Also**

<code>writeLegendListForm</code>	<code>zoomToEnvelope</code>	<code>queryForm</code>	<code>printIt</code>
<code>sendMapXML</code>	<code>fullExtent</code>	<code>setupGeocode</code>	<code>writeOptionForm</code>
<code>writeModeFrame</code>	<code>zoomBack</code>	<code>getStoredQueries</code>	<code>extractIt</code>
<code>writeModeLayers</code>	<code>showLayer</code>	<code>findForm</code>	<code>getLegend</code>
<code>deleteClick</code>	<code>update MeasureBox</code>	<code>checkSelected</code>	<code>hasLayer</code>
<code>hidelayer</code>	<code>resetClick</code>	<code>writeBufferForm</code>	

## **clipLayer**

### **Description:**

Clips the visible area of the style sheet/layer containing the main map display.

Uses: isNav4.

Calls: getLayer in aimsDHTML.js.

Called by: createLayer, putExtentOnOVMMap, and boxIt in aimsDHTML.js, by panMouse in aimsNavigation.js, and by processXML in aimsXML.js.

### **Category:**

Extended Map

### **File:**

aimsDHTML.js

### **Syntax:**

clipLayer(name, clipleft, cliptop, clipright, clipbottom)

### **Arguments:**

name	String containing name of style sheet/layer.
clipleft	Numeric representing pixel x-coordinate of left edge of clip.
cliptop	Numeric representing pixel y-coordinate of top edge of clip.
clipright	Numeric representing pixel x-coordinate of right edge of clip.
clipbottom	Numeric representing pixel y-coordinate of bottom edge of clip.

### **Returned Value:**

None

### **See Also**

createLayer	putExtentOnOVMMap
boxIt	panMouse
processXML	

## **convertHexToDec**

**Description:**

Converts an HTML-style RGB hexadecimal color number string to its decimal equivalent.

**Category:**

Basic Map

**File:**

aimsCommon.js

**Syntax:**

convertHexToDec(hexColor)

**Arguments:**

hexColor      String containing hexadecimal color string

**Returned Value:**

String      String representing the converted value in decimal RGB format ( e.g. “255, 0, 0”).

## convertUnits

### Description:

Converts a distance value of one unit type into the distance value in another unit type. The available unit types are: METERS, FEET, MILES, and KILOMETERS.

Called by: getScaleBarDistance and calcDistance in aimsMap.js.

### Category:

Basic Map

### File:

aimsMap.js

### Syntax:

convertUnits(theDist1,mUnits,sUnits)

### Arguments:

theDist      Numeric representing original distance value in mUnits.

mUnits      String containing unit type of theDist value.

sUnits      String containing unit type that theDist value will be converted to.

### Returned Value:

Numeric      Convert distance value in sUnits unit type.

### See Also

[getScaleBarDistance](#)

[calcDistance](#)

## **createLayer**

### **Description:**

Creates a new style sheet/layer in the MapFrame page. These are used to enable the interactivity of the map page.

Uses: isNav4.

Called in: MapFrame.htm in setting up various elements in the page.

### **Category:**

Extended Map

### **File:**

aimsDHTML.js

### **Syntax:**

createLayer(name, left, top, width, height, visible, content)

### **Arguments:**

name	String containing name of new style sheet/layer.
left	Numeric representing pixel x-coordinate of left edge of name.
top	Numeric representing pixel y-coordinate of top edge of name.
width	Numeric representing width of name.
height	Numeric representing height of name.
visible	Boolean indicating initial state of the visibility of name.
content	String containing the content of name. This string constructs the style sheet/layer.

### **Returned Value:**

None

## **customMapTool**

### **Description:**

Allows the developer to add code for a custom tool. Any toolModes > 1000 are available for use. Developers must also update useCustomFunction to reflect any custom tools desired.

Called by: mapTool in aimsClick.js if toolMode is > 1000.

### **Category:**

Custom

### **File:**

aimsCustom.js

### **Syntax:**

customMapTool(e)

### **Arguments:**

e Event passed by browser.

### **Returned Value:**

None

### **See Also**

mapTool

## **deleteClick**

### **Description:**

Deletes the last click location from the collection of click locations.

Sets:selectCount to zero. Updates the clickPointX, clickPointY, and clickMeasure arrays.

Updates: clickCount and totalMeasure.

Uses: noOverlay.

Calls: sendToServer and writeXML.

### **Category:**

Interactive Map

### **File:**

aimsClick.js

### **Syntax:**

`deleteClick()`

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

[writeXML](#)

[sendToServer](#)

## displayAttributeData

### Description:

Parses the returned ArcXML response from an IDIdentify, Select, or Query request and displays attribute data in an HTML table.

Calls: setLayerFields, getXMLErrorMessage, parseRecordString, clearLeadingSpace, getFieldNames, getFieldValues, getIDValue, getEnvelopeXYs, justGetFeatureCount, saveLastExtent, sendMapXML, and hideRetrieveData.

Uses: ActiveLayerIndex, selectCount, showSelectedData, useExternalWindow, useTextFrame, toolMode, useFieldAlias, queryStartRecord, XMLEndPos, textFrameBackColor, tableBackColor, zoomToSingleSelect, selectPointMargin, selectMargin, and maxFeaturesReturned.

Uses: selectLeft, selectRight, selectTop, selectBottom, hyperLinkLayers, hyperLinkFields, LayerName, selectPoints, AliasFieldName, LayerFields, LayerFieldType, and aliasFieldAlias arrays.

Called by processXML in aimsXML.js.

### Category:

IDIdentify/Hyperlink

### File:

aimsIDIdentify.js

### Syntax:

displayAttributeData(theReply)

### Arguments:

theReply      String containing returned ArcXML response from query.

### Returned Value:

None

### See Also

See the next page.

## **displayAttributeData (continued)**

### **See Also**

processXML	hideRetrieveData	setLayerFields	justGetFeature Count
getIDValue	getEnvelopeXYs	getXMLErrorMessage	saveLast Extent
getFieldNames	getFieldValues	parseRecordString	clearLeadingSpace

## **extractIt**

### **Description:**

Used for processing Extract requests. Not implemented.

Calls: hideLayer.

Called by: clickFunction in aimsClick.js.

### **Category:**

Custom

### **File:**

aimsCustom.js

### **Syntax:**

extractIt()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

clickFunction    hideLayer

## **findForm**

**Description:**

Displays the HTML form for the Find Mode.

Uses: ActiveLayerIndex and useTextFrame.

Calls: setLayerFields in aimsLayers.js.

Called by: clickFunction in aimsClick.js.

**Category:**

Query

**File:**

aimsQuery.js

**Syntax:**

findForm()

**Arguments:**

None

**Returned Value:**

None

**See Also**

clickFunction    setLayerFields

## **fixSingleQuotes**

### **Description:**

Replaces single quotes with double single quotes in strings. This function sets up interior single quotes and apostrophes in strings sent to the server for queries.

Called by: sendQueryString and parseFieldSamples in aimsQuery.js.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

fixSingleQuotes(inputString)

### **Arguments:**

inputString      String to be processed.

### **Returned Value:**

String      Converted string.

### **See Also**

sendQueryString      parseFieldSamples

## **fullExtent**

### **Description:**

Sets the main map display extent to the defined full extent.

Uses: aimsDHTMLPresent, hspc, vspc, left, top, right, bottom, fullLeft, fullTop, fullRight, and fullBottom.

Calls: moveLayer in aimsDHTML.js, saveLastExtent in aimsMap.js, and sendMapXML in aimsXML.js.

Called by: clickFunction in aimsClick.js.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

fullExtent()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

moveLayer

saveLastExtent

sendMapXML

clickFunction

## **getAllFieldValues**

### **Description:**

Parses out the values of one field from an ArcXML query response.

Uses: xmlEndPos.

Calls: justGetFieldValue in aimsCommon.js.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

getAllFieldValues(theReply,theField,recCount)

### **Arguments:**

theReply      String containing ArcXML query response to be parsed.

theField      String containing name of field to be used.

recCount      Numeric representing number of records to be parsed.

### **Returned Value:**

Array      List of values parsed.

### **See Also**

[justGetFieldValue](#)

## getBufferAttributeData

### Description:

Parses returned ArcXML response from Buffer request and displays attribute data in an HTML table.

Calls: setLayerFields, getXMLErrorMessage, parseRecordString, clearLeadingSpace, justGetFeatureCount, getFieldNames, getFieldValues, getIDValue, and hideRetrieveData.

Uses : ActiveLayerIndex, selectCount, showSelectedData, useExternalWindow, useTextFrame, toolMode, useFieldAlias, queryStartRecord, maxFeaturesReturned, bufferTargetLayerIndex, XmlEndPos, textFrameBackColor, tableBackColor, ActiveLayer, ActiveLayerType, bufferTargetLayer, useFieldAlias, and showHyper.

Uses: hyperLinkLayers, hyperLinkFields, LayerName, selectPoints, AliasFieldName, and LayerFields, LayerFieldType, and AliasFieldNames arrays.

Called by: processXML in aimsXML.js.

### Category:

Buffer

### File:

aimsBuffer.js

### Syntax:

getBufferAttributeData(theReply)

### Arguments:

theReply      String containing ArcXML buffer response to be parsed.

### Returned Value:

None.

### See Also

processXML

hideRetrieveData

setLayerFields justGetFeatureCount

getXMLErrorMessage

parseRecordString

getIDValue

clearLeadingSpace

getFieldNames

getFieldValues

## **getCommandLineParams**

### **Description:**

Parses the command line parameters, if any, for viewer settings.

Uses: imsURL, imsOVURL, startLeft, startTop, startRight, startBottom, limitLeft, limitTop, limitRight, limitBottom, getStartingExtent, getLimitExtent, imsQueryURL, serverURL, imsGeocodeURL, and canLoad.

Called by: checkParams in ArcIMSParam.js.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

getCommandLineParams(cmdString)

### **Arguments:**

cmdString      String containing command line parameters.

### **Returned Value:**

None

### **See Also**

checkParams

## getEnvelopeXYs

### Description:

Parses the returned ArcXML response for envelope coordinates.

Uses: dQuote and XmlEndPos.

Called by: displayAttributeData in aimsIDentify.js, getLayers in aimsLayers.js, and getXYs and getOVXYs in aimsXML.js.

### Category:

Basic Map

### File:

aimsXML.js

### Syntax:

getEnvelopeXYs(theString, startpos)

### Arguments:

theString      String containing returned ArcXML response.

startpos      Numeric representing starting character position in theString to start parsing.

### Returned Value:

Array      List containing minx, miny, maxx, and maxy.

### See Also

displayAttributeData    getLayers

getXYs      getOVXYs

## **getFieldNames**

### **Description:**

Parses the returned ArcXML query response for field names.

Called by: `getBufferAttributeData` in `aimsBuffer.js` and `displayAttributeData` in `aimsIDeIdentify.js`.

### **Category:**

Basic Map

### **File:**

`aimsCommon.js`

### **Syntax:**

`getFieldNames(recordString)`

### **Arguments:**

`recordString` String containing returned ArcXML response.

### **Returned Value:**

Array List containing field names.

### **See Also**

`getBufferAttributeData`      `displayAttributeData`

## **getFieldValues**

### **Description:**

Parses the returned ArcXML query response for field values.

Uses: ActiveLayerIndex and ActiveLayerType.

Uses: LayerShapeField array.

Called by: getBufferAttributeData in aimsBuffer.js and displayAttributeData in aimsIDentify.js.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

getFieldValues(recordString)

### **Arguments:**

recordString    String containing returned ArcXML response.

### **Returned Value:**

Array        List containing field values.

### **See Also**

getBufferAttributeData        displayAttributeData

## **getFind**

### **Description:**

Sets up the Find request to be sent to the server.

Uses: ActiveLayerIndex and ActiveLayerType.

Uses: LayerIDField, LayerShapeField, LayerFields, LayerFieldType, showBuffer, and setQueryString arrays.

Calls: showRetrieveData in aimsMap.js, makeXMLsafe in aimsCommon.js, send to server in aimsXML.js, and writeFindRequest in aimsQuery.js and writeFindRequest in aimsQuery.js.

Called by: Find Form.

### **Category:**

Query

### **File:**

aimsQuery.js

### **Syntax:**

getFind(theValue)

### **Arguments:**

theValue      String containing value to be matched.

### **Returned Value:**

None

### **See Also**

showRetrieveData      makeXMLsafe

writeFindRequest

## **getGeocodeLayers**

### **Description:**

Requests a list of layers configured for geocoding.

Uses: imsGeocodeURL.

Calls: sendToServer in aimsXML.js.

### **Category:**

Geocode

### **File:**

aimsGeocode.js

### **Syntax:**

getGeocodeLayers()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

sendToServer

## **getGeocodeParams**

### **Description:**

Requests a list of geocoding parameters.

Uses: imsGeocodeURL.

Calls: sendToServer in aimsXML.js.

Called by: setupGeocode in aimsGeocode.js

### **Category:**

Geocode

### **File:**

aimsGeocode.js

### **Syntax:**

getGeocodeParams()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

sendToServer      setupGeocode

## **getIDValue**

### **Description:**

Returns the value for the active layer's ID field.

Uses: ActiveLayerIndex.

Uses: LayerIDField array.

Called by: displayAttributeData in aimsIDIdentify.js and by getBufferAttributeData in aimsBuffer.js.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

getIDValue(fieldNameArray, fieldValueArray)

### **Arguments:**

fieldNameArray      List of field names.

fieldValueArray      List of field values.

### **Returned Value:**

String      Value of ID field.

### **See Also**

displayAttributeData    getBufferAttributeData

## **getImageXY**

### **Description:**

Translates click in MapFrame page units into image pixel coordinates.

Sets: mouseX and mouseY.

Uses: isNav, hspc, and vspc.

Called by: mapTool in aimsClick.js, getMouse, startZoomBox, startZoomOutBox, and startPan in aimsNavigation.js, and by select and startSelectBox in aimsSelect.js.

### **Category:**

Interactive Map

### **File:**

aimsNavigation.js

### **Syntax:**

getImageXY(e)

### **Arguments:**

e                  Event passed by browser.

### **Returned Value:**

None

### **See Also**

mapTool	getMouse	startZoomBox
select	startPan	startZoomOutBox
startSelectBox		

# getInsideString

## Description:

Extracts an interior string from another string.

Called by: `getURL` and `getLegendURL` in `aimsXML.js`.

## Category:

Basic Map

## File:

`aimsCommon.js`

## Syntax:

```
getInsideString(inString,beforeString,afterString,startpos,limitpos,caseSensitive)
```

## Arguments:

<code>inString</code>	String to be used to extract interior string.
<code>beforeString</code>	String preceding starting position of interior string.
<code>afterString</code>	String following ending position of interior string.
<code>startpos</code>	Numeric representing character position to begin search for <code>beforeString</code> . Use zero to search from beginning position.
<code>limitpos</code>	Numeric representing character position that <code>beforeString</code> must precede. If no limit is imposed, zero is used.
<code>caseSensitive</code>	Boolean indicating if search will be case sensitive.

## Returned Value:

`String`      Interior string

## See Also

`getURL`      `getLegendURL`

## **getLayer**

### **Description:**

Gets the style sheet (Netscape layer) referenced by name.

Uses: isNav4, isIE, and isNav.

Called by: isVisible, moveLayer, setLayerBackgroundColor, hideLayer, showLayer, clipLayer and replaceLayerContent in aimsDHTML.js.

### **Category:**

Extended Map

### **File:**

aimsDHTML.js

### **Syntax:**

getLayer(name)

### **Arguments:**

name            String containing name of style sheet/layer.

### **Returned Value:**

Object            style sheet/layer referenced by name.

### **See Also**

isVisible            moveLayer            setLayerBackgroundColor

hideLayer            showLayer            clipLayer

replaceLayerContent

## getLayerFieldNames

### Description:

Gets a list of field names for the referenced layer.

Uses: LayerFieldList array.

Called by: showLayerInfo in aimsLayers.js.

### Category:

Extended Map

### File:

aimsLayers.js

### Syntax:

getLayerFieldNames(layerIndex)

### Arguments:

layerIndex      Numeric representing index number of layer, starting with topmost layer as zero.

### Returned Value:

Array      List of field names for the referenced layer.

### See Also

showLayerInfo      getLayerFieldPrecisions

getLayerFieldSizes      getLayerFieldTypes

## **getLayerFieldPrecisions**

### **Description:**

Gets a list of field precisions for the referenced layer.

Uses: LayerFieldList and LayerFieldPrecisionList arrays.

Called by: showLayerInfo in aimsLayers.js.

### **Category:**

Extended Map

### **File:**

aimsLayers.js

### **Syntax:**

getLayerFieldPrecisions (layerIndex)

### **Arguments:**

layerIndex      Numeric representing index number of layer, starting with topmost layer as zero.

### **Returned Value:**

Array      List of field precisions for the referenced layer.

### **See Also**

showLayerInfo      getLayerFieldNames

getLayerFieldSizes      getLayerFieldTypes

## **getLayerFieldSizes**

### **Description:**

Gets a list of field sizes for the referenced layer.

Uses: LayerFieldList and LayerFieldSizList arrays.

Called by: showLayerInfo in aimsLayers.js.

### **Category:**

Extended Map

### **File:**

aimsLayers.js

### **Syntax:**

getLayerFieldSizes (layerIndex)

### **Arguments:**

layerIndex      Numeric representing index number of layer, starting with topmost layer as zero.

### **Returned Value:**

Array      List of field sizes for the referenced layer.

### **See Also**

[showLayerInfo](#)      [getLayerFieldNames](#)

[getLayerFieldPrecisions](#)      [getLayerFieldTypes](#)

## **getLayerFieldTypes**

### **Description:**

Gets a list of field types for the referenced layer.

Uses: LayerFieldList and LayerFieldTypeList arrays.

Called by: showLayerInfo in aimsLayers.js.

### **Category:**

Extended Map

### **File:**

aimsLayers.js

### **Syntax:**

getLayerFieldTypes (layerIndex)

### **Arguments:**

layerIndex      Numeric representing index number of layer, starting with topmost layer as zero.

### **Returned Value:**

Array      List of field types for the referenced layer.

### **See Also**

showLayerInfo                    getLayerFieldNames

getLayerFieldPrecisions        getLayerFieldSizes

## getLayers

### Description:

Gets a list of layer settings for the current MapService.

Uses: layerCount and xmlEndPos.

Sets: ActiveLayerIndex, ActiveLayer, ActiveLayerType, and canQuery.

Sets: LayerName, LayerType, LayerVisible, LayerExtent, LayerIsFeature, LayerID, LayerIDField, LayerShapeField, LayerMinScale, LayerMaxScale, LayerFieldTypeList, LayerFieldList, LayerRenderString, LayerFieldSizeList, LayerFieldPrecisionList, ClassRenderLayer, and LayerRenderString arrays.

Calls: getEnvelopeXYs in aimsXML.js.

Called by: processStartExtent in aimsCommon.js.

### Category:

Extended Map

### File:

aimsLayers.js

### Syntax:

getLayers(theReply)

### Arguments:

theReply      String containing returned ArcXML GET\_SERVICE\_INFO response.

### Returned Value:

None

### See Also

getEnvelopeXYs      processStartExtent

## **getLegend**

### **Description:**

Sends an ArcXML request to create a legend image.

Sets: legendVisible and drawLegendOnly.

Calls: showRetrieveMap in aimsMap.js, and writeXML and sendToServer in aimsXML.js.

Called by: clickFunction in aimsClick.js, and by processXML and getURL in aimsXML.js.

### **Category:**

Legend

### **File:**

aimsLegend.js

### **Syntax:**

getLegend()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

[clickFunction](#)

[processXML](#)

[getURL](#)

## **getLegendURL**

### **Description:**

Extracts the URL for a legend image from a returned ArcXML response.

Uses: dQuote

Calls: getInsideString in aimsCommon.js

Called by: processXML and getURL in aimsXML.js.

### **Category:**

Basic Map

### **File:**

aimsXML.js

### **Syntax:**

getLegendURL(theReply)

### **Arguments:**

theReply      String containing returned ArcXML response.

### **Returned Value:**

String      URL of legend image.

### **See Also**

processXML      getURL

## **getMapHeight**

**Description:**

Returns height of the MapFrame page.

**Category:**

Basic Map

**File:**

aimsCommon.js

**Syntax:**

getMapHeight()

**Arguments:**

None

**Returned Value:**

Numeric      MapFrame height

## **getMapWidth**

**Description:**

Returns width of the MapFrame page.

**Category:**

Basic Map

**File:**

aimsCommon.js

**Syntax:**

getMapWidth()

**Arguments:**

None

**Returned Value:**

Numeric      MapFrame width

## **getMapXY**

### **Description:**

Translates click on map image to map coordinates.

Uses: xDistance, yDistance, iWidth, iHeight, left, and bottom.

Sets: mouseX, mouseY, pixelX, pixelY, mapX, and mapY.

### **Category:**

Interactive Map

### **File:**

aimsNavigation.js

### **Syntax:**

getMapXY(xIn,yIn)

### **Arguments:**

xIn            Numeric representing image pixel x-coordinate.

yIn            Numeric representing image pixel y-coordinate.

### **Returned Value:**

None

## **getMoreData**

### **Description:**

Requests more records of the current selection set of features.

Sets: queryStartRecord.

Uses: setQueryString, selectionMode, clickType, imsQueryURL, and selectXMLMode.

Calls: writeQueryXML in aimsQuery.js, writeGetFeatures3 and writeShapeSelect in aimsSelect.js, showRetrieveData in aimsMap.js, and sendToServer in aimsXML.js.

Called by: displayAttributeData in aimsIDIdentify.js.

### **Category:**

Select

### **File:**

aimsSelect.js

### **Syntax:**

getMoreData(startRecord)

### **Arguments:**

startRecord     Numeric representing the record number of the selected features that the returned group of records will start from.

### **Returned Value:**

None

### **See Also**

[displayAttributeData](#)

## getMouse

### Description:

Gets the coordinates at the mouse position and parses them to the function for the current cursor mode.

Uses: isIE, hasOVMMap, ovIsVisible, ovMapIsLayer, mouseX, mouseY, i2Width, i2Height, iWidth, iHeight, zooming, selectBox, panning, x2, y2, pixelX, pixelY, xDistance, yDistance, left, bottom, mapX, mapY, toolMode, showXYs, showScalePercent, orBoxSize, numDecimals, and mapScaleFactor.

Calls: getImageXY, setClip, and panMouse in aimsNavigation.js, chkMouseUp in aimsClick.js, and calcDistance in aimsMap.js.

Called by: browser onmousemove event defined in checkParams in ArcIMSpParam.js.

### Category:

Interactive Map

### File:

aimsNavigation.js

### Syntax:

getMouse(e)

### Arguments:

e              Event from browser

### Returned Value:

None

### See Also

[getImageXY](#)

[setClip](#)

[checkParams](#)

[chkMouseUp](#)

[calcDistance](#)

[panMouse](#)

## **getOVImageXY**

### **Description:**

Translates click in MapFrame page into image pixel coordinates.

Sets: mouseX and mouseY.

Uses: isNav, mouseX, mouseY, ovHspc, and ovVspc.

Called by: ovMap2Click in aimsNavigation.js.

### **Category:**

Interactive Map

### **File:**

aimsNavigation.js

### **Syntax:**

getImageXY(e)

### **Arguments:**

e              Event passed by browser.

### **Returned Value:**

None

### **See Also**

[ovMap2Click](#)

## getOVXYs

### Description:

Extracts extent coordinates from returned ArcXML response.

Sets: fullOVLeft, fullOVBottom, fullOVRRight, fullOVTop, fullOVWidth, and fullOVHeight.

Calls: getEnvelopeXYs in aimsXML.js.

Called by: processXML in aimsXML.js.

### Category:

Basic Map

### File:

aimsXML.js

### Syntax:

getOVXYs(theString)

### Arguments:

theString      String containing ArcXML response with extent coordinates for the overview map display.

### Returned Value:

None

### See Also

getEnvelopeXYs      processXML

## **getPath**

### **Description:**

Extracts the path (without the filename) from a full URL.

Called by: checkParams in ArcIMSPparam.js.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

getPath(theFullPath)

### **Arguments:**

theFullPath      String containing full URL.

### **Returned Value:**

String      Path portion of URL, without filename.

### **See Also**

checkParams

## **getPrintLegend**

### **Description:**

Adds a legend image to the print routine.

Uses: printLegURL.

Calls: writePrintPage in aimsPrint.js.

Called by: processXML in aimsXML.js.

### **Category:**

Print

### **File:**

aimsPrint.js

### **Syntax:**

getPrintLegend()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

[processXML](#)

[writePrintPage](#)

## **getPrintMap**

### **Description:**

Starts the print routine.

Uses: iWidth, iHeight, legVis2, legendVisible, aimsLegendPresent, and imsURL.

Calls: showRetrieveMap in aimsMap.js, and writeXML and sendToServer in aimsXML.js.

Called by: Print form.

Sets: printTitle.

### **Category:**

Print

### **File:**

aimsPrint.js

### **Syntax:**

getPrintMap(title)

### **Arguments:**

title            String containing title to be used on Print page.

### **Returned Value:**

None

### **See Also**

[showRetrieveMap](#)      [writeXML](#)

[sendToServer](#)

## **getPrintOV**

### **Description:**

Adds an overview map image to the print routine.

Uses: i2Width, i2Height, drawOVExtentBox, and imsOVURL.

Calls: writeOVXML and sendToServer in aimsXML.js.

Called by: Print form.

### **Category:**

Print

### **File:**

aimsPrint.js

### **Syntax:**

getPrintOV()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

[writeOVXML](#)

[sendToServer](#)

## **getScaleBarDistance**

### **Description:**

Calculates the distance that should be displayed in the ScaleBar. Value is based on current extent and ScaleBarUnits.

Uses: MapUnits, left, right, bottom, top, and ScaleBarUnits.

Calls: convertUnits in aimsMap.js.

Called by: writeXML in aimsXML.js.

Sets: ScaleBarPrecision.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

getScaleBarDistance()

### **Arguments:**

None

### **Returned Value:**

Numeric      Distance to be used in the ScaleBar.

### **See Also**

convertUnits      writeXML

## **getStartExtent**

### **Description:**

Creates ArcXML request string to obtain limit extent. If extent is already known, request is not sent to server.

Uses: getLimitExtent, hasOVMap, imsOVURL, imsURL, and XMLMode.

Calls: sendToServer or processXML in aimsXML.js.

Called by: startUp in aimsCommon.js.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

getStartExtent()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

sendToServer              processXML

startUp

## **getStoredQueries**

### **Description:**

Creates ArcXML request string to obtain the StoredQueries in the MapService. Request is sent to server.

Uses: imsQueryURL.

Calls: sendToServer in aimsXML.js.

Called by: clickFunction in aimsClick.js.

### **Category:**

Query

### **File:**

aimsQuery.js

### **Syntax:**

getStoredQueries()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

sendToServer

clickFunction

## **getURL**

### **Description:**

Extracts the URL for a map image from ArcXML response. Also sets various globals. If a legend image is required, calls getLegendURL for URL.

Uses: dQuote.

Calls: getInsideString in aimsCommon.js and getLegendURL in aimsXML.js.

Called by: processXML in aimsXML.js.

Sets: theImageURLPath, theImageType, noOverlay, theImagePath, theServiceName, and legendImage.

### **Category:**

Basic Map

### **File:**

aimsXML.js

### **Syntax:**

getURL(theReply)

### **Arguments:**

theReply      String containing returned ArcXML response.

### **Returned Value:**

String      URL of map image.

### **See Also**

getInsideString      getLegendURL

processXML

## **getXMLErrorMessage**

### **Description:**

Extracts the error message returned in an ArcXML response, if any.

Called by: processXML in aimsXML.js, getBufferAttributeData in aimsBuffer.js, and displayAttributeData in aimsIDentify.js.

### **Category:**

Basic Map

### **File:**

aimsXML.js

### **Syntax:**

getXMLErrorMessage(theString)

### **Arguments:**

theString      String containing returned ArcXML response.

### **Returned Value:**

String      Error message. If none, an empty string is returned.

### **See Also**

processXML      getBufferAttributeData

displayAttributeData

## **getXYs**

### **Description:**

Extracts map extent coordinates from returned ArcXML response.

Uses: panFactor, iWidth, aimsLayersPresent, mapScaleFactor, hasTOC, legendVisible, appDir and LayerListOpen.

Calls: getEnvelopeXYs in aimsXML.js and writeLayerListForm in aimsLayers.js.

Called by: processStartExtent in aimsCommon.js and processXML in aimsXML.js.

Sets: xDistance, yDistance, xHalf, yHalf, panX, and panY.

### **Category:**

Basic Map

### **File:**

aimsXML.js

### **Syntax:**

getXYs(theString)

### **Arguments:**

theString      String containing ArcXML response with extent coordinates for the Overview Map display.

### **Returned Value:**

None

### **See Also**

getEnvelopeXYs      processXML

processStartExtent      writeLayerListForm

## **hasLayer**

### **Description:**

Tests if a style sheet (Netscape layer) exists.

Uses: isNav4, isIE, and isNav.

Called by: checkParams in ArcIMSParam.js, clickFunction in aimsClick.js, showRetrieveData, hideRetrieveData, showRetrieveMap, and hideRetrieveMap in aimsMap.js, stopPan and panMouse in aimsNavigation.js, and processXML in aimsXML.js.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

hasLayer(name)

### **Arguments:**

name        String containing name of style sheet/layer.

### **Returned Value:**

Boolean       If True, name exists.

### **See Also**

checkParams	clickFunction	showRetrieveData
hideRetrieveData	showRetrieveMap	hideRetrieveMap
stopPan	panMouse	processXML

## hideLayer

**Description:**

“Hides” (set the visibility to false) a Style Sheet (Netscape layer).

Uses: isNav4.

Calls: getLayer in aimsDHTML.js.

Called by: bufferIt im aimsBuffer.js, clickFunction in aimsClick.js, useCustomFunction im aimsCustom.js, toggleOVMap in aimsDHTML.js, hideRetrieveData and hideRetrieveMap in aimsMap.js, stopZoomBox, stopZoomOutBox, and stopPan in aimsNavigation.js, printIt in aimsPrint.js, and stopSelectBox in aimsSelect.js.

**Category:**

Extended Map

**File:**

aimsDHTML.js

**Syntax:**

hideLayer(name)

**Arguments:**

name            String containing name of style sheet/layer.

**Returned Value:**

None

**See Also**

bufferIt	clickFunction	useCustomFunction
toggleOVMap	hideRetrieveData	hideRetrieveMap
stopZoomBox	stopZoomOutBox	stopPan
printIt		stopSelectBox

## **hideRetrieveData**

### **Description:**

“Hides” (set the visibility to false) of the “Loading Data” graphic that is displayed when the viewer is retrieving data from the server.

Calls: hideLayer in aimsDHTML.js and hasLayer in aimsMap.js.

Called by: getBufferAttributeData in aimsBuffer.js, by processStartExtent in aimsCommon.js, by displayAttributeData and parseHyperLink in aimsIDeIdentify.js, and by sendToServer in aimsXML.js.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

hideRetrieveData()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

hideLayer	hasLayer	getBufferAttributeData
processStartExtent	displayAttributeData	parseHyperLink
sendToServer		

## **hideRetrieveMap**

### **Description:**

“Hides” (set the visibility to false) the “Loading Map” graphic that is displayed when the viewer is retrieving a new map image from the server.

Calls: hideLayer in aimsDHTML.js and hasLayer in aimsMap.js.

Called by: writePrintPage in aimsPrint.js, and sendToServer and processXML in aimsXML.js.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

hideRetrieveMap ()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

hideLayer

hasLayer

writePrintPage

sendToServer

## hyperLink

### Description:

Sends an ArcXML request for data to execute a hyperlink.

Uses: currentHyperLinkLayer, highlightedOne, mouseX, mouseY, searchTolerance, xDistance, iWidth, pixelTolerance, mapX, mapY, swapSelectFields, ActiveLayerIndex, imsQueryURL, hyperlinkXMLMode, currentHyperLinkField, and selectFields.

Uses: the LayerIDField and LayerShapeField arrays.

Calls: writeGetFeatures in aimsSelect.js, showRetrieveData in aimsMap.js, getMapXY in aimsNavigation.js., and sendToServer in aimsXML.js.

Called by: clickFunction in aimsClick.js.

### Category:

IDeIdentify/Hyperlink

### File:

aimsIDeIdentify.js

### Syntax:

hyperLink(e)

### Arguments:

e              Event from browser.

### Returned Value:

None

### See Also

writeSetFeatures	getMapXY
showRetrieveData	sendToServer
clickFunction	

## **identify**

### **Description:**

Sends an ArcXML request for data to execute a hyperlink.

Uses: currentHyperLinkLayer, highlightedOne, mouseX, mouseY, searchTolerance, xDistance, iWidth, pixelTolerance, mapX, mapY, imsQueryURL, identifyXMLMode, and selectFields.

Calls: writeGetFeatures in aimsSelect.js, showRetrieveData in aimsMap.js, getMapXY in aimsNavigation.js, and sendToServer in aimsXML.js.

Called by: clickFunction in aimsClick.js.

### **Category:**

IDIdentify/Hyperlink

### **File:**

aimsIDIdentify.js

### **Syntax:**

identify(e)

### **Arguments:**

e              Event from browser.

### **Returned Value:**

None

### **See Also**

writeSetFeatures        showRetrieveData

sendToServer        clickFunction

## **isVisible**

### **Description:**

Returns the visibility of the style sheet (Netscape layer).

Uses: isNav and isIE.

Calls: getLayer in aimsDHTML.js.

### **Category:**

Extended Map

### **File:**

aimsDHTML.js

### **Syntax:**

isVisible(name)

### **Arguments:**

name            String containing name of style sheet/layer.

### **Returned Value:**

Boolean        If True, name is visible.

### **See Also**

[getLayer](#)

## **justGetFeatureCount**

### **Description:**

Returns the number of features returned in ArcXML query response.

Uses: dQuote.

Calls: justGetValue in aimsCommon.js

Called by: getBufferAttributeData in aimsBuffer.js, displayAttributeData and parseHyperLink in aimsIDeIdentify.js, and parseFieldSamples in aimsQuery.js.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

justGetFeatureCount(theReply)

### **Arguments:**

theReply      String containing returned ArcXML response.

### **Returned Value:**

Numeric      Number of features returned.

### **See Also**

justGetValue      getBufferAttributeData

displayAttributeData      parseHyperLink

parseFieldSamples

## **justGetFieldValue**

### **Description:**

Returns a single value from a returned ArcXML query response.

Uses: dQuote.

Calls: justGetValue in aimsCommon.js.

Called by: getAllFieldValues in aimsCommon.js.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

justGetFieldValue(theReply,theField,startpos).

### **Arguments:**

theReply      String containing returned ArcXML response.

theField      String containing name of field to obtain value from.

startpos      Numeric representing character position to begin search.

### **Returned Value:**

String      Field value.

### **See Also**

[justGetValue](#)

## **justGetMap**

### **Description:**

Function that sends an ArcXML request for a map image.

Uses: left, top, right, bottom, and debugOn.

Calls: writeXML and sendToServer in aimsXML.js.

### **Category:**

Basic Map

### **File:**

aimsXML.js

### **Syntax:**

justGetMap(theURL, extentLeft, extentTop, extentRight, extentBottom, getOVMMap)

### **Arguments:**

theURL	String containing URL to MapService.
extentLeft	Numeric representing minimum map x-coordinate.
extentTop	Numeric representing minimum map x-coordinate.
extentRight	Numeric representing minimum map x-coordinate.
extentBottom	Numeric representing minimum map x-coordinate.
getOVMMap	Boolean indicating if an overview map image should also be requested. If True, a second request will be sent for the overview map image and will not bypass normal viewer map image processing.

### **Returned Value:**

None

### **See Also**

[writeXML](#)    [sendToServer](#)

## **justGetValue**

### **Description:**

Returns an interior string value from a returned ArcXML query response.

Called by: hystGetFieldValue and justGetFeatureCount in aimsCommon.js.

Sets: xmlEndPos.

### **Category:**

BasicMap

### **File:**

aimsCommon.js

### **Syntax:**

justGetValue(theReply,preString,postString,startpos)

### **Arguments:**

theReply	String containing returned ArcXML response.
preString	String containing preceding value string. Value string will begin after the last character of preString.
postString	String containing string following value string. The last character of the value string will be just before the first character of postString.
startpos	Numeric representing character position to begin search.

### **Returned Value:**

String      Value string.

### **See Also**

[justGetFieldValue](#)      [justGetFeatureCount](#)

## **makeXMLsafe**

### **Description:**

Function that converts string into an XMLcompatible string. Offending characters are replaced by HTML entities that transmit properly. Used in setting up query strings to be sent to server.

Calls: swapStuff in aimsCommon.js.

Called by: sendQueryString, getFind, and sendStoredQuery in aimsQuery.js.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

makeXMLsafe(oldString)

### **Arguments:**

oldString      String to be converted.

### **Returned Value:**

String      Converted string.

### **See Also**

swapStuff

sendQueryString

getFind

sendStoredQuery

## **mapTool**

### **Description:**

Function that executes on click of cursor within the MapFrame map display.

Calls: function associated with current toolMode.

Called by: browser event defined in ArcIMSparam.js.

### **Category:**

Interactive Map

### **File:**

aimsClick.js

### **Syntax:**

mapTool(e)

### **Arguments:**

e              Event from browser.

### **Returned Value:**

None

### **See Also**

[clickFunction](#)

## **moveLayer**

### **Description:**

Function that moves Style Sheet (Netscape layer) to new position on page.

Uses: isNav4.

Calls: getLayer in aimsDHTML.js.

Called by: putExtentOnOVMMap and boxIt in aimsDHTML.js, fullExtent, startExtent, and zoomBack in aimsMap.js, startZoomBox, startZoomOutBox, startPan, and panMouse in aimsNavigation.js, and startSelectBox in aimsSelect.js, and processXML in aimsXML.js.

### **Category:**

Extended Map

### **File:**

aimsDHTML.js

### **Syntax:**

moveLayer(name, x, y)

### **Arguments:**

name        String containing name of style sheet/layer.

x        Numeric representing new upper-left pixel x-coordinate.

y        Numeric representing new upper-left pixel y-coordinate.

### **Returned Value:**

None

### **See Also**

getLayer

startPan

putExtentOnOVMMap

panMouse

boxIt

startSelectBox

fullExtent

processXML

startExtent

zoomBack

startZoomBox

startZoomOutBox

## **numberorder**

### **Description:**

Function to be used in an array numeric sort.

Called by: sort() method of an array.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

numberorder

### **Arguments:**

a,b           Arbitrary elements used for sorting purposes. These are for construction purposes and are not used in the actual function call.

### **Returned Value:**

Numeric       Value of array element comparison.

## **ovMap2Click**

### **Description:**

Function that executes on click in overview map display.

Uses: mouseY, mouseX, ovBorderWidth, zooming, panning, and selectBox.

Calls: getOVIImageXY and ovMapClick in aimsNavigation.js.

Called by: onmousedown event of overview map image defined in MapFrame.htm.

### **Category:**

Interactive Map

### **File:**

aimsNavigation.js

### **Syntax:**

ovMap2Click(e)

### **Arguments:**

e              Event from browser.

### **Returned Value:**

None

### **See Also**

[getOVIImageXY](#)

[ovMapClick](#)

## ovMapClick

### Description:

Function that defines a new extent for the main map display using overview map image coordinates. The passed coordinates are converted to map coordinates and are used as the center point of the new extent. A request for a new map image is sent to the server.

Uses: i2Width, i2Height, fullOVWidth, fullOVHeight, fullOVLeft, fullOVBottom, left, top, right, and bottom.

Calls: saveLastExtent in aimsMap.js and sendMapXML in aimsXML.js.

Called by: ovMap2Click in aimsNavigation.js.

### Category:

Interactive Map

### File:

aimsNavigation.js

### Syntax:

ovMapClick(x,y)

### Arguments:

x              Numeric representing image x-coordinate.

y              Numeric representing image y-coordinate.

### Returned Value:

None

### See Also

saveLastExtent        sendMapXML

ovMap2Click

# **pan**

## **Description:**

Function that pans main map extent to be centered around click on map display. Used as an alternative to the DHTML “sliding” pan of startPan and stopPan in aimsNavigation.js.

Uses: mouseX, mouseY, lastLeft, lastTop, lastRight, lastBottom, left, top, right, and bottom.

Calls: getMapXY in aimsNavigation, saveLastExtent in aimsMap.js, and sendMapXML in aimsXML.js.

## **Category:**

Interactive Map

## **File:**

aimsNavigation.js

## **Syntax:**

pan(e)

## **Arguments:**

e              Event from browser.

## **Returned Value:**

None

## **See Also**

[getMapXY](#)

[saveLastExtent](#)

[sendMapXML](#)

## **panButton**

### **Description:**

Pans main map extent in the desired direction. Pan distance is set by global variables panX and panY using panFactor.

Uses: left, top, right, bottom, xDistance, yDistance, and panFactor.

Calls: saveLastExtent and checkFullExtent in aimsMap.js, and sendMapXML in aimsXML.js.

Sets: panX and panY.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

panButton(panType)

### **Arguments:**

panType      Numeric representing direction number. 1=West, 2=North, 3=East, 4=South, 5=Southwest, 6=Northwest, 7=Northeast, and 8=Southeast.

### **Returned Value:**

None

### **See Also**

checkFullExtent      sendMapXML

saveLastExtent

## **panMouse**

### **Description:**

Pans main map display with movement of mouse.

Uses: x1, y1, x2, y2, iWidth, iHeight, hspc, and vspc.

Calls: hasLayer in aimsMap.js and clipLayer and moveLayer in aimsDHTML.js.

Called by: getMouse in aimsNavigation.js.

### **Category:**

Interactive Map

### **File:**

aimsNavigation.js

### **Syntax:**

panMouse()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

hasLayer      clipLayer

moveLayer      getMouse

## **parseEntity**

### **Description:**

Replaces common HTML entities with the characters they represent.

Calls: swapStuff in aimsCommon.js.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

parseEntity(oldString)

### **Arguments:**

oldString      String containing entities to be parsed.

### **Returned Value:**

String      Converted string.

### **See Also**

[swapStuff](#)

## **parseFieldSamples**

### **Description:**

Parses out a list of sample field values from a returned ArcXML query response.

Sets: selectData array.

Calls: justGetFeatureCount and clearLeadingSpace in aimsCommon.js.

Called by: processXML in aimsXML.js.

### **Category:**

Query

### **File:**

aimsQuery.js

### **Syntax:**

parseFieldSamples(theReply)

### **Arguments:**

theReply      String containing returned ArcXML response.

### **Returned Value:**

None

### **See Also**

justGetFeatureCount      clearLeadingSpace

processXML

## **parseGeocodeLayers**

### **Description:**

Parses out lists of geocoding layer parameters from a returned ArcXML response.

Uses: dQuote.

Called by: processXML in aimsXML.js.

Sets: GCLayerCount.

Sets: GCLayers, GCLayerID, and GCLayerStyle arrays.

### **Category:**

Geocode

### **File:**

aimsGeocode.js

### **Syntax:**

parseGeocodeLayers(theReply)

### **Arguments:**

theReply      String containing returned ArcXML response.

### **Returned Value:**

None

### **See Also**

processXML

## **parseGeocodeParams**

### **Description:**

Parses out basic geocode parameters for a specific layer from a returned ArcXML response.

Uses: dQuote.

Called by: processXML in aimsXML.js.

Sets: GCidCount.

Sets: GCid, GClabel, and GCdesc arrays.

### **Category:**

Geocode

### **File:**

aimsGeocode.js

### **Syntax:**

parseGeocodeParams(theReply,theLayer)

### **Arguments:**

theReply      String containing returned ArcXML response.

theLayer      String containing name of layer.

### **Returned Value:**

Boolean      If True, parameters are set for layer.

### **See Also**

[processXML](#)

## **parseGeocodeResults**

### **Description:**

Parses out results from a returned ArcXML geocode response and displays the results in a window.

Uses: dQuote and TextFrame.

Sets: GCpointCount, showGeocode, geocodeX, geocodeY, geocodeLabel, and useExternalWindow.

Sets: GCscore, GCaddress, GCpointX, and GCpointY arrays.

Calls: sendMapXML in aimsXML.js.

Called by: processXML in aimsXML.js.

### **Category:**

Geocode

### **File:**

aimsGeocode.js

### **Syntax:**

parseGeocodeResults(theReply)

### **Arguments:**

theReply      String containing returned ArcXML response.

### **Returned Value:**

None

### **See Also**

sendMapXML      processXML

## **parseHyperLink**

### **Description:**

Parses out the URL in a field value from a returned ArcXML geocode response.

Uses: dQuote, hyperlinkWindowWidth, hyperlinkWindowHeight, currentHyperLinkField, newSelectCount, ActiveLayerIndex, and debugOn.

Uses: LayerName array.

Calls: justGetFeatureCount and untag in aimsCommon.js, and hideRetrieveData in aimsMap.js.

Called by: processXML in aimsXML.js.

### **Category:**

IDeIdentify/Hyperlink

### **File:**

aimsIDeIdentify.js

### **Syntax:**

parseHyperLink(theReply)

### **Arguments:**

theReply      String containing returned ArcXML response.

### **Returned Value:**

None

### **See Also**

justGetFeatureCount	untag
hideRetrieveData	processXML

## **parseRecordString**

### **Description:**

Parses out record data from a returned ArcXML query response.

Sets: xmlEndPos.

Called by: displayAttributeData in aimsIDeIdentify.js and by getBufferAttributeData in aimsBuffer.js.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

parseRecordString(theReply,startpos)

### **Arguments:**

theReply      String containing returned ArcXML response.

startpos      Numeric representing starting character position of search.

### **Returned Value:**

String      Record string.

### **See Also**

displayAttributeData      getBufferAttributeData

## **parseStoredQueries**

### **Description:**

Parses out StoredQuery parameters from a returned ArcXML query response.

Uses: ActiveLayer, and dQuote.

Uses: storedQueryName, storedQueryString, storedQueryVariable, storedQueryVarCount, and storedQueryFieldList arrays.

Calls: storedQueryForm in aimsQuery.js.

Called by: processXML in aimsXML.js.

Sets: storedQueryCount and storedQueryIndex.

### **Category:**

Query

### **File:**

aimsQuery.js

### **Syntax:**

parseStoredQueries(theReply)

### **Arguments:**

theReply      String containing returned ArcXML response.

### **Returned Value:**

None

### **See Also**

storedQueryForm      processXML

## **printIt**

### **Description:**

Starts the Print procedure.

Uses: useTextFrame.

Calls: hideLayer in aimsDHTML.js.

Called by: clickFunction in aimsClick.js.

### **Category:**

Print

### **File:**

aimsPrint.js

### **Syntax:**

printIt()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

hideLayer      clickFunction

## processStartExtent

### Description:

Processes the starting extent and sets up layers (if aimsLayers.js is loaded).

Uses: getStartingExtent, aimsLayersPresent, aimsQueryPresent, useStoredQuery, chkUnits, hasOVMap, hasTOC, showTOC, aimsGeocodePresent, GCLayerCount, hasToolBarOnLayer, isNav, aimsDHTMLPresent, and enforceFullExtent.

Calls: getXYs, writeBlankMapXML, and sendMapXML in aimsXML.js, swapStuff in aimsCommon.js, hideRetrieveData in aimsMap.js, getLayers in aimsLayers.js, checkStoredQueries in aimsQuery.js, replaceLayerContent and toggleOVMap in aimsDHTML.js, and custom function getLayerListContent.

Called by: processXML in aimsXML.js.

Sets: left, top, right, bottom, startLeft, startTop, startRight, startBottom, xDistance, yDistance, iWidth, iHeight, mapScaleFactor, xHalf, yHalf, panX, panY, panFactor, MapUnits, mouseX, mouseY, pixelX, pixelY, mapX, mapY, lastLeft, lastTop, lastRight, lastBottom, , fullLeft, fullTop, fullRight, fullBottom, fullWidth, fullHeight, useGeocode, useReverseGeocode, sQuote, ovIsVisible.

### Category:

Basic Map

### File:

aimsCommon.js

### Syntax:

processStartExtent(theReply)

### Arguments:

theReply      String containing returned ArcXML response.

### Returned Value:

None

### See Also

see the next page

## **processStartExtent (continued)**

### **See Also**

getXYs	toggleOVMap	writeBlankMapXML	swapStuff
processXML	sendMapXML	getLayers	checkStoredQueries
replaceLayerContent	hideRetrieveData		

## processXML

### Description:

Passes the ArcXML response to the appropriate function for processing. The key global is XMLMode, which determines which function to call.

Uses: printMapURL, printLegURL, printOVURL, legendImage, toolMode, debugOn, legendVisible, noOverlay, hasOVMap, ovVisible, pastStart, getBufferData, aimsGeocodePresent, useGeocode, useReverseGeocode, drawLegendOnly, imsOVURL, imsURL, hspc, vspc, iWidth, iHeight, aimsClickPresent, clickCount, aimsBufferPresent, getLimitExtent, imsGeocodeURL, GSActiveLayer, appDir, fullLeft, fullRight, fullTop, fullBottom, left, right, top, bottom, limitLeft, limitRight, limitTop, limitBottom, fullOVLeft, fullOVRight, fullOVTop, fullOVBottom, fullWidth, fullHeight, fullOVWidth, fullOVHeight, useExternalWindow, useTextFrame, imageLimitLeft, imageLimitRight, imageLimitTop, and, imageLimitBottom.

Uses: GCLayers array.

Calls: appropriate function to parse ArcXML response.

Called by: passXML in dynamic connector-created page in PostFrame on loading.

### Category:

Basic Map

### File:

aimsXML.js

### Syntax:

processXML(theReplyIn)

### Arguments:

theReplyIn      String containing returned ArcXML response.

### Returned Value:

None

### See Also

See the next page

## processXML (continued)

### See Also

replacePlus	getXYs	parseIDFieldData
unescape	afterMapRefresh	updateMeasureBox
getXMLErrorMessage	moveLayer	showLegend
getURL	clipLayer	writeOverlayXML
getOVXYs	hasLayer	showRetrieveMap
getLegendURL	displayAttributeData	sendToServer
getPrintOV	getBufferAttributeData	processRoute
getPrintLegend	parseFieldSamples	parseHyperLink
writePrintPage	writeQueryForm	parseGeocodeLayers
sendMapXML	parseStoredQueries	parseGeocodeParams
parseGeocodeResults	useCustomFunction	writeOVXML
putExtentOnOVMap	hideRetrieveMap	writeGetBufferData
processStartExtent	processCatalog	processGeocode
processReverseGeocode	processProx	processRoute

## **putExtentOnOVMap**

### **Description:**

Updates extent box in overview map display.

Uses: fullOVWidth, fullOVHeight, i2Width, i2Height, fullOVLeft, fullOVTop, left, top, right, bottom, ovBorderWidth, isNav5up, ovExtentboxSize, and cornerOffset.

Calls: moveLayer and clipLayer in aimsDHTML.js.

Called by: processXML in aimsXML.js and toggleOVMap in aimsDHTML.js.

### **Category:**

Extended Map

### **File:**

aimsDHTML.js

### **Syntax:**

`putExtentOnOVMap()`

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

`moveLayer`

`clipLayer`

`processXML`

`toggleOVMap`

## **queryForm**

### **Description:**

Sets up the Query Form.

Uses: showSampleValues, ActiveLayerIndex, fieldIndex, and imsQueryURL.

Uses: LayerFields array.

Calls: setLayerFields in aimsLayers.js, writeFieldSample and writeQueryForm in aimsQuery.js, and sendToServer in aimsXML.js.

Called by: setActiveLayer in aimsLayers.js and clickFunction is aimsClick.js.

### **Category:**

Query

### **File:**

aimsQuery.js

### **Syntax:**

queryForm()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

setLayerFields	writeFieldSample
writeQueryForm	sendToServer
setActiveLayer	clickFunction

## **reloadApp**

### **Description:**

Reloads the viewer if the browser is Netscape. Netscape's resize event causes the various frames to try to refresh. This refresh is not total, occasionally causing the viewer to stall.

Uses: isNav.

Called by: document event defined in viewer.htm.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

reloadApp()

### **Arguments:**

None

### **Returned Value:**

None

## **replaceLayerContent**

### **Description:**

Replaces the content of a style sheet (Netscape layer).

Uses: isNav4 and isIE.

Calls: getLayer in aimsDHTML.js.

Called by: processStartExtent and writeModeLayers in aimsCommon.js.

### **Category:**

Extended Map

### **File:**

aimsDHTML.js

### **Syntax:**

replaceLayerContent(name, content)

### **Arguments:**

name        String containing name of style sheet/layer.

content      String containing new content for style sheet/layer.

### **Returned Value:**

None

### **See Also**

getLayer      processStartExtent

writeModeLayers

## **replacePlus**

### **Description:**

Replaces plus signs (“+”) with spaces in a string that has been HTML encoded by Java.

Called by: processXML in aimsXML.js.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

replacePlus(inText)

### **Arguments:**

inText            String containing text to be converted.

### **Returned Value:**

String            Converted string.

### **See Also**

processXML

## **resetClick**

### **Description:**

Resets collection of click locations to zero.

Uses: noOverlay, legendVisible, imsURL, blankImage, and toolMode.

Calls: sendToServer, writeXML, and pdateMeasurebox.

Called by: clickFunction in aimsClick.js.

Sets: selectCount, lastToMeasure, clickCount, and totalMeasure to zero. Resets the clickPointX, clickPointY, and clickMeasure arrays.

### **Category:**

Interactive Map

### **File:**

aimsClick.js

### **Syntax:**

resetClick()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

clickFunction	sendToServer
updateMeasureBox	writeXML

## **resetError**

**Description:**

Resets JavaScript error checking to default.

Use: clearError to disable error checking.

**Category:**

Basic Map

**File:**

aimsCommon.js

**Syntax:**

resetError()

**Arguments:**

None

**Returned Value:**

Boolean      Returns False.

**See Also**

clearError

## **saveLastExtent**

### **Description:**

Function that saves the last map extent.

Uses: left, top, right, and bottom.

Called by: displayAttributeData in aimsIDeIdentify.js, fullExtent, startExtent, zoomToPoint, zoomToEnvelope, zoomScale, zoomButton, and panButton in aimsMap.js, ovMapClick, zoomin, zoomout, stopZoomBox, stopZoomOutBox, stopPan, and pan in aimsNavigation.js, and showHighlight in aimsSelect.js.

Sets: lastLeft, lastRight, lastTop, and lastBottom.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

saveLastExtent()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

displayAttributeData	stopZoomBox	fullExtent
stopZoomOutBox	startExtent	stopPan
zoomToPoint	pan	zoomToEnvelope
showHighlight	zoomScale	zoomButton
panButton	ovMapClick	zoomin
zoomout		

## **select**

### **Description:**

Starts spatial query of active layer using a single point.

Uses: mouseX, mouseY, mapX, mapY, xDistance, yDistance, and pixelTolerance.

Calls: getImageXY and getMapXY in aimsNavigation.js, showRetrieveData in aimsMap.js, writeGetFeatures2 in aimsSelect.js, and sendToServer in aimsXML.js.

Called by: mapTool in aimsClick.js.

Sets: searchTolerance, queryStartRecord, selectEnvelope, selectMode=2, imsQueryURL, and selectXMLMode.

### **Category:**

Basic Map

### **File:**

aimsSelect.js

### **Syntax:**

select(e)

### **Arguments:**

e                  Event from browser.

### **Returned Value:**

None

### **See Also**

getImageXY	getMapXY
showRetrieveData	writeGetFeatures2
sendToServer	mapTool

## **sendCustomToServer**

### **Description:**

Sends a custom query request to the QueryServer.

Sets: form values in PostFrame page.

Calls: sendToServer in aimsXML.js.

### **Category:**

Basic Map

### **File:**

aimsXML.js

### **Syntax:**

sendCustomToServer(XMLRequest, theFunction, theType)

### **Arguments:**

XMLRequest String containing returned ArcXML response.

theFunction String containing function name that will handle response.

theType Numeric representing value to be passed to XMLMode.

### **Returned Value:**

None

### **See Also**

[sendToServer](#)

## sendMapXML

### Description:

Sends an ArcXML request to obtain a new map image to the MapServer.

Uses: debugOn and imsURL.

Calls: beforeMapRefresh and showRetrieveMap in aimsMap.js and writeXML and sendToServer in aimsXML.js.

Called by: bufferIt in aimsBuffer.js, clickFunction in aimsClick.js, processStartExtent in aimsCommon.js, parseGeocodeResults in aimsGeocode, displayAttributeData in aimsIDeIdentify.js, fullExtent, startExtent, zoomBack, zoomToPoint, zoomToEnvelope, zoomScale, zoomButton, and panButton in aimsMap.js, ovMapClick, zoomin, zoomout, stopZoomBox, stopZoomOutBox, stopPan, and pan in aimsNavigation.js, showHighlight in aimsSelect.js, and process XML in aimsXML.js.

### Category:

Basic Map

### File:

aimsXML.js

### Syntax:

sendMapXML()

### Arguments:

None

### Returned Value:

None

### See Also

beforeMapRefresh	showRetrieveMap	displayAttributeData	zoomToPoint
writeXML	sendToServer	fullExtent	zoomToEnvelope
bufferIt	clickFunction	startExtent	zoomScale
processStartExtent	parseGeocodeResults	zoomBack	zoomButton
panButton	ovMapClick	zoomin	zoomout
stopZoomBox	stopZoomOutBox	stopPan	pan
showHighlight	processXML		

## **sendQueryString**

### **Description:**

Sends a standard SQL query request to the Query Server.

Uses: setQueryString, selectionMode, ActiveLayerIndex, LayerFieldCount, showBuffer, imsQueryURL, and queryXMLMode.

Uses: selectData, LayerFields, LayerIDField, and LayerFieldType arrays.

Calls: fixSingleQuotes, swapQuotes, and makeXMLsafe in aimsCommon.js, showRetrieveData in aimsMap.js, writeQueryXML in aimsQuery.js, and sendToServer in aimsXML.js.

Called by: query form created by writeQueryForm in aimsQuery.js.

### **Category:**

Query

### **File:**

aimsQuery.js

### **Syntax:**

sendQueryString(newString)

### **Arguments:**

newString      String containing expression to be used in where clause of a standard SQL query.

### **Returned Value:**

None

### **See Also**

fixSingleQuotes	swapQuotes
makeXMLsafe	writeQueryXML
sendToServer	showRetrieveData

## **sendShapeSelect**

### **Description:**

Starts spatial query request using user-created shape.

Uses: queryStartRecord, selectionMode, showBuffer, clickCount, imsQueryURL, and selectXMLMode.

Uses: clickPointX, clickPointY arrays.

Calls: showRetrieveData in aimsMap.js writeShapeSelect in aimsSelect.js, and sendToServer in aimsXML.js.

Called by: Shape form in select.htm.

### **Category:**

Select

### **File:**

aimsSelect.js

### **Syntax:**

sendShapeSelect(theType)

### **Arguments:**

theType      Numeric representing type of shape used (1=Line; 2=Polygon).

### **Returned Value:**

None

### **See Also**

showRetrieveData      writeShapeSelect

sendToServer

## **sendStoredQuery**

### **Description:**

Constructs the query string for a stored query and sends it to the Query Server.

Uses: imsQueryURL and queryXMLMode.

Uses: storedQueryString, storedQueryVarCount, and storedQueryVariable arrays.

Calls: showRetrieveData in aimsMap.js, swapStuff, and makeXMLsafe in aimsCommon.js, and sendToServer in aimsXML.js.

Called by: writeStoredQueryForm in aimsQuery.js and the form it creates.

Sets: setQueryString.

### **Category:**

Query

### **File:**

aimsQuery.js

### **Syntax:**

sendStoredQuery(theIndex,theValue)

### **Arguments:**

theIndex      Numeric representing index of element in storeQueryString,  
                  storedQueryVarCount, and storedQueryVariable arrays.

theValue      String containing user-input value(s) for query. If multiple values are used,  
                  the values are separated by a bar (“|”).

### **Returned Value:**

None

### **See Also**

showRetrieveData	swapStuff
makeXMLsafe	sendToServer
writeStoredQueryForm	

## **sendToServer**

### **Description:**

Used to pass ArcXML requests on to the server(s). This function updates the form in the PostFrame page with current request, processing function name and location of form template files (header.htm and footer.htm).

Uses: okToSend, XMLMode, and debugOn.

Calls: showRetrieveMap, hideRetrieveMap, and hideRetrieveData in aimsMap.js.

Called by: functions sending ArcXML requests to the server(s).

### **Category:**

Basic Map

### **File:**

aimsXML.js

### **Syntax:**

sendToServer(URLString,XMLRequest,theType)

### **Arguments:**

URLString      String containing MapService URL for request.

XMLRequest      String containing ArcXML request.

theType      Numeric representing mode number to be passed to XMLMode. Used by processXML in passing ArcXML response to appropriate function for processing.

### **Returned Value:**

None

### **See Also**

showRetreiveMap      hideRetrieveMap      hideRetrieveData

## **setActiveLayer**

### **Description:**

Sets the active layer.

Uses: queryOpen, useExternalWindow, tableBackColor, and useTextFrame.

Uses: LayerID, LayerType, and LayerName arrays.

Calls: setLayerFields in aimsLayers.js and queryForm in aimsQuery.js.

Called by: form in toc.htm.

Sets: fieldIndex, selectCount, showBuffer, ActiveLayerIndex, ActiveLayer, and ActiveLayerType.

### **Category:**

Extended Map

### **File:**

aimsLayers.js

### **Syntax:**

setActiveLayer(i)

### **Arguments:**

i      Numeric representing index of layer. Index of topmost layer is zero.

### **Returned Value:**

None

### **See Also**

setLayerFields      queryForm

## **setClip**

### **Description:**

Clips the zoom box layer to the mouse coordinates.

Uses: x1, y1, x2, y2, zleft, zright, ztop, zbottom, and ovBoxSize.

Calls: boxIt and clipLayer in aimsDHTML.js.

Called by: getMouse in aimsNavigation.js.

### **Category:**

Interactive Map

### **File:**

aimsNavigation.js

### **Syntax:**

setClip()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

boxIt              clipLayer

getMouse

## **setExtent**

### **Description:**

Sets current map extent. No request for new map image is sent.

Uses: left, top, right, and bottom.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

```
setExtent(newLeft, newTop, newRight, newBottom)
```

### **Arguments:**

newLeft	Numeric representing new extent left x-coordinate.
newTop	Numeric representing new extent top y-coordinate.
newRight	Numeric representing new extent right x-coordinate.
newBottom	Numeric representing new extent bottom y-coordinate.

### **Returned Value:**

None

## **setFullExtent**

### **Description:**

Sets the full extent.

Used to set initial full extent. Does not update any limit globals.

Uses: fullLeft, fullTop, fullRight, fullBottom, fullWidth, and fullHeight.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

```
setFullExtent(maxLeft, maxTop, maxRight, maxBottom)
```

### **Arguments:**

maxLeft	Numeric representing new maximum extent left x-coordinate.
maxTop	Numeric representing new maximum extent top y-coordinate.
maxRight	Numeric representing new maximum extent right x-coordinate.
maxBottom	Numeric representing new maximum extent bottom y-coordinate.

### **Returned Value:**

None

## **setLayerBackgroundColor**

### **Description:**

Sets the background color of the style sheet (Netscape layer).

Uses: isNav4.

Calls: getLayer in aimsDHTML.js.

Called by: setZoomColor in aimsNavigation.js and style sheet/layer setup in MapFrame.htm.

### **Category:**

Extended Map

### **File:**

aimsDHTML.js

### **Syntax:**

setLayerBackgroundColor(name, color)

### **Arguments:**

name        String containing name of style sheet/layer.

color        String containing HTML color name or number in hexadecimal RGB format (#RRGGBB).

### **Returned Value:**

None

### **See Also**

getLayer      setZoomColor

## **setLayerFields**

### **Description:**

Sets parameters in the field arrays using fields from the current active layer.

Uses: swapSelectFields, useFieldAlias.

Uses: LayerFields, LayerFieldType, selFieldList, LayerFieldList, AliasFieldName, LayerShapeField, AliasFieldAlias, LayerName, and fieldAliasList arrays.

Called by: getBufferAttributeData in aimsBuffer.js, displayAttributeData in aimsIDentify.js, setActiveLayer in aimsLayers.js, queryForm and findForm in aimsQuery.js.

Sets: selectFields, LayerFieldCount, and canQuery

### **Category:**

Extended Map

### **File:**

aimsLayers.js

### **Syntax:**

setLayerFields(layerIndex)

### **Arguments:**

layerIndex      Numeric representing index of layer. Index of topmost layer is zero.

### **Returned Value:**

None

### **See Also**

getBufferAttributeData      displayAttributeData

setActiveLayer      queryForm

findForm

## **setupGeocode**

### **Description:**

Starts the Geocoding mode.

Calls: getGeocodeParams in aimsGeocode.js.

Called by: clickFunction in aimsClick.js.

### **Category:**

Geocode

### **File:**

aimsGeocode.js

### **Syntax:**

setupGeocode()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

[getGeocodeParams](#)    [clickFunction](#)

## **setZoomColor**

### **Description:**

Sets the zoom box color using the global zoomBoxColor.

Uses: zoomBoxColor.

Calls: setLayerBackgroundColor in aimsDHTML.js.

### **Category:**

Interactive Map

### **File:**

aimsNavigation.js

### **Syntax:**

setZoomColor()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

[setLayerBackgroundColor](#)

## **showHighlight**

### **Description:**

Sends an ArcXML request for new map image, highlighting one selected feature and zooming the main map display to an area surrounding the specified feature.

Uses: ActiveLayerIndex, fullWidth, fullHeight, selectPointMargin, selectMargin, xDistance, yDistance, legendVisible, selectType and legendTemp.

Uses: LayerIDField, selectPoints, selectLeft, selectTop, selectRight, and selectBottom arrays

Calls: sendMapXML in aimsXML.js and saveLastExtent in aimMap.js.

Called by: hyperlink in data display created by displayAttributeData in aimsIDIdentify.js.

Sets: highlightedOne, left, top, right, and bottom.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

showHighlight(selNum)

### **Arguments:**

selNum      Numeric representing element index in the selectPoints, selectLeft, selectTop, selectRight, and selectBottom arrays. These arrays hold values and envelope coordinates from currently selected features of the active layer.

### **Returned Value:**

None

### **See Also**

sendMapXML      saveLastExtent  
displayAttributeData

## **showLayer**

### **Description:**

Shows (set the visibility to true) a style sheet (Netscape layer).

Uses: isNav4.

Calls: getLayer in aimsDHTML.js.

Called by: clickFunction and updateMeasureBox in aimsClick.js, toggleOVMap in aimsDHTML.js, showRetrieveData and showRetrieveMap in aimsMap.js, startZoomBox, startZoomOutBox, and startPan in aimsNavigation.js, startSelectBox and clearSelection in aimsSelect.js, processXML in aimsXML.js.

### **Category:**

Extended Map

### **File:**

aimsDHTML.js

### **Syntax:**

showLayer(name)

### **Arguments:**

name            String containing name of style sheet/layer.

### **Returned Value:**

None

### **See Also**

clickFunction	updateMeasureBox
toggleOVMap	showRetrieveData
showRetrieveMap	startZoomBox
startZoomOutBox	startPan
startSelectBox	clearSelection
processXML	

## showLayerInfo

### Description:

Displays layer information.

Uses: useExternalWindow, useTextFrame, textFrameBackColor, and tableBackColor.

Uses: LayerIsFeature, LayerName, LayerID, LayerExtent, LayerScale, LayerMaxScale, and LayerType arrays.

Calls: getLayerFieldNames, getLayerFieldTypes, getLayerFieldSizes, and getLayerFieldPrecisions in aimsLayers.js.

Called by: functions and form in toc.htm.

### Category:

Extended Map

### File:

aimsLayers.js

### Syntax:

showLayerInfo(layerIndex)

### Arguments:

layerIndex      Numeric representing index of layer. Index of topmost layer is zero.

### Returned Value:

None

### See Also

getLayerFieldNames	getLayerFieldTypes
getLayerFieldSizes	getLayerFieldPrecisions

## **showLegend**

### **Description:**

Displays a graphic legend image.

Uses: hasTOC and legendImage.

Called by: processXML in aimsXML.js.

### **Category:**

Legend

### **File:**

aimsLegend.js

### **Syntax:**

showLegend()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

[processXML](#)

## showRetrieveData

### Description:

Displays an animated GIF image while awaiting the response to a data request to the server.

Calls: hasLayer in aimsMap.js and showLayer in aimsDHTML.js.

Called by: startMap in aimsCommon.js, identify and hyperlink in aimsIDeIdentify.js, sendQueryString, getFind, and sendStoredQuery in aimsQuery.js, select, sendShapeSelect, and getMoreData in aimsSelect.js.

### Category:

Basic Map

### File:

aimsMap.js

### Syntax:

showRetrieveData()

### Arguments:

None

### Returned Value:

None

### See Also

startMap	identify
hyperlink	sendQueryString
sendStoredQuery	select
sendShapeSelect	getMoreData
getFind	hasLayer
showLayer	

## **showRetrieveMap**

### **Description:**

Displays an animated GIFimage while awaiting the reponse to a map request to the server.

Calls: hasLayer in aimsMap.js and showLayer in aimsDHTML.js.

Called by: getLegend in aimsLegend.js, by getPrintMap in aimsPrint.js, by sendToServer, sendMapXML, and processXML in aimsXML.js.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

showRetrieveMap()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

getLegend      hasLayer

getPrintMap    sendToServer

sendMapXML showLayer

processXML

## **startExtent**

### **Description:**

Sets map extent to starting extent and requests a new map image.

Uses: aimsDHTMLPresent, hspc, vspc, left, top, right, and bottom.

Calls: moveLayer in aimsDHTML.js, saveLastExtent in aimsMap.js, and sendMapXML in aimsXML.js.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

startExtent()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

moveLayer      saveLastExtent

sendMapXML

## **startMap**

### **Description:**

Starts the MapService loading procedure. Checks if aimsGeneric.js is loaded.

Uses: aimsGenericPresent and catURL.

Calls: showRetrieveData in aimsMap.js, startUp in aimsCommon.js, and sendToServer in aimsXML.js.

Called by: checkParams in ArcIMSPparam.js.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

startMap()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

showRetrieveData      startUp

sendToServer      checkParams

## **startPan**

### **Description:**

Begins the interactive Pan mode.

Uses: hspc, vspc, mouseX, mouseY, iWidth, iHeight, panning, x1, y1, x2, and y2.

Calls: moveLayer in aimsDHTML.js, and stopPan and getImageXY in aimsNavigation.js.

Called by: mapTool in aimsClick.js.

### **Category:**

Interactive Map

### **File:**

aimsNavigation.js

### **Syntax:**

startPan(e)

### **Arguments:**

e              Event from browser.

### **Returned Value:**

None

### **See Also**

moveLayer              stopPan

getImageXY              mapTool

## **startSelectBox**

### **Description:**

Begins the interactive Select by Rectangle mode.

Uses: hspc, vspc, mouseX, mouseY, iWidth, iHeight, selectBox, x1, y1, x2, y2, zleft, zright, ztop, zbottom, and highlightedOne.

Calls: moveLayer, showLayer, and boxIt in aimsDHTML.js, getImageXY in aimsNavigation.js, and stopSelectBox in aimsSelect.js.

Called by: mapTool in aimsClick.js.

### **Category:**

Select

### **File:**

aimsSelect.js

### **Syntax:**

startSelectBox(e)

### **Arguments:**

e              Event from browser.

### **Returned Value:**

None

### **See Also**

moveLayer              showLayer              boxIt              getImageXY

stopSelectBox              mapTool              clipLayer

## **startUp**

### **Description:**

Gets map image size and starts loading MapService if the global imsURL has a value.

Uses: imsURL and imsOVURL.

Calls: getStartExtent in aimsCommon.js.

Called by: startMap in aimsCommon.js.

Sets: iWidth, iHeight, and toggleOVVisible.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

startUp()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

[getStartExtent](#)      [startMap](#)

## **startZoomBox**

### **Description:**

Begins the interactive Zoom In mode.

Uses: hspc, vspc, mouseX, mouseY, iWidth, iHeight, zooming, x1, y1, x2, y2, zleft, zright, ztop, and zbottom.

Calls: moveLayer, showLayer, clipLayer, and boxIt in aimsDHTML.js, and getImageXY, and stopZoomBox in aimsNavigation.js.

Called by: mapTool in aimsClick.js.

### **Category:**

Interactive Map

### **File:**

aimsNavigation.js

### **Syntax:**

startZoomBox(e)

### **Arguments:**

e                  Event from browser.

### **Returned Value:**

None

### **See Also**

moveLayer

showLayer

boxIt

getImageXY

stopZoomBox

mapTool

clipLayer

## **startZoomOutBox**

### **Description:**

Begins the interactive Zoom Out mode.

Uses: hspc, vspc, mouseX, mouseY, iWidth, iHeight, zooming, x1, y1, x2, y2, zleft, zright, ztop, and zbottom.

Calls: moveLayer, showLayer, clipLayer, and boxIt in aimsDHTML.js, and getImageXY and stopZoomOutBox in aimsNavigation.js.

Called by: mapTool in aimsClick.js.

### **Category:**

Interactive Map

### **File:**

aimsNavigation.js

### **Syntax:**

startZoomBox(e)

### **Arguments:**

e              Event from browser.

### **Returned Value:**

None

### **See Also**

moveLayer              showLayer              boxIt              getImageXY

stopZoomOutBox        mapTool              clipLayer

## **stopPan**

### **Description:**

Stops the interactive Pan mode and requests a new map image.

Uses: panning, lastLeft, lastRight, lastTop, lastBottom, x1, y1, x2, y2, pixelX, pixelY, ztop, xDistance, yDistance, iWidth, iHeight, left, top, right, bottom, and blankImage.

Calls: saveLastExtent, checkFullExtent, and hasLayer in aimsMap.js, sendMapXML in aimsXML.js, and hideLayer in aimsDHTML.js.

Called by: startPan in aimsNavigation.js and mapTool in aimsClick.js.

### **Category:**

Interactive Map

### **File:**

aimsNavigation.js

### **Syntax:**

stopPan(e)

### **Arguments:**

e              Event from browser.

### **Returned Value:**

None

### **See Also**

saveLastExtent

checkFullExtent

hasLayer

startPan

hideLayer

mapTool

sendMapXML

## **stopSelectBox**

### **Description:**

Stops the interactive Select by Rectangle mode and sends requests to the ImageServer and QueryServer.

Uses: selectBox, lastLeft, lastRight, lastTop, lastBottom, mapX, mapY, mouseX, mouseY, pixelTolerance, pixelX, pixelY, xDistance, yDistance, iWidth, iHeight, left, top, right, bottom, zleft, ztop, zright, zbottom, queryStartRecord, selectEnvelope, drawSelectBoundary, selectionMode, showBuffer, imsQueryURL, and selectXMLMode.

Calls: getMapXY in aimsNavigation.js, writeGetFeatures2 in aimsSelect.js, hideLayer in aimsDHTML.js, and sendToServer in aimsXML.js.

Called by: startSelectBox in aimsSelect.js and mapTool in aimsClick.js.

### **Category:**

Select

### **File:**

aimsSelect.js

### **Syntax:**

stopSelectBox(e)

### **Arguments:**

e              Event from browser.

### **Returned Value:**

None

### **See Also**

getMapXY	writeGetFeatures2
hideLayer	startSelectBox
mapTool	sendToServer

## **stopZoomBox**

### **Description:**

Stops the interactive Zoom In mode and requests a new map image.

Uses: zooming, lastLeft, lastRight, lastTop, lastBottom, pixelX, pixelY, xDistance, yDistance, iWidth, iHeight, left, top, right, bottom, zleft, ztop, zright, and zbottom.

Calls: zoomin and saveLastExtent in aimsMap.js, hideLayer in aimsDHTML.js, and sendMapXML in aimsXML.js.

Called by: startZoomBox in aimsNavigation.js and mapTool in aimsClick.js.

### **Category:**

Interactive Map

### **File:**

aimsNavigation.js

### **Syntax:**

stopZoomBox(e)

### **Arguments:**

e              Event from browser.

### **Returned Value:**

None

### **See Also**

zoomin	saveLastExtent
hideLayer	sendMapXML
mapTool	startZoomBox

## **stopZoomOutBox**

### **Description:**

Stops the interactive Zoom Out mode and requests a new map image.

Uses: zooming, lastLeft, lastRight, lastTop, lastBottom, xDistance, yDistance, fullWidth, fullLeft, fullRight, fullTop, fullBottom, iWidth, iHeight, left, top, right, bottom, zleft, ztop, zright, zbottom, and enforceFullExtent.

Calls: zoomout and saveLastExtent in aimsMap.js, hideLayer in aimsDHTML.js, and sendMapXML in aimsXML.js.

Called by: startZoomOutBox in aimsNavigation.js and mapTool in aimsClick.js.

### **Category:**

Interactive Map

### **File:**

aimsNavigation.js

### **Syntax:**

stopZoomBox(e)

### **Arguments:**

e              Event from browser.

### **Returned Value:**

None

### **See Also**

zoomout	saveLastExtent
hideLayer	sendMapXML
mapTool	startZoomOutBox

## **storedQueryForm**

### **Description:**

Starts the StoredQuery procedure.

Uses: storedQueryIndex.

Calls: writeStoredQueryForm in aimsQuery.js.

Called by: parseStoredQueries in aimsQuery.js.

### **Category:**

Query

### **File:**

aimsQuery.js

### **Syntax:**

storedQueryForm()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

[writeStoredQueryForm](#)      [parseStoredQueries](#)

## **swapQuotes**

### **Description:**

Replaces double quotes with single quotes in a string.

Called by: sendQueryString in aimsQuery.js.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

swapQuotes(inText)

### **Arguments:**

inText            String to be converted.

### **Returned Value:**

String            Converted string.

### **See Also**

[sendQueryString](#)

## **swapStuff**

### **Description:**

Replaces an interior string with another string.

Called by: processStartExtent, parseEntity, and makeXMLsafe in aimsCommon.js and sendStoredQuery in aimsQuery.js.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

swapStuff(oldString,oldStuff,newStuff)

### **Arguments:**

oldString      String to be converted.

oldStuff      String found within oldString.

newStuff      String that will replace oldStuff in oldString.

### **Returned Value:**

String      Converted string.

### **See Also**

processStartExtent      parseEntity

makeXMLsafe      sendStoredQuery

## **tempGetSamples**

### **Description:**

Sends an ArcXML request for sample field values for the Query form to the QueryServer.

Uses: useTextFrame textFrameColor, ActiveLayerIndex, numberDataSamples, and imsQueryURL.

Calls: writeFieldSample in aimsQuery.js and sendToServer in aimsXML.js.

Called by: query form created by writeQueryForm in aimsQuery.js.

Sets: showSampleValues.

### **Category:**

Query

### **File:**

aimsQuery.js

### **Syntax:**

tempGetSamples(theField)

### **Arguments:**

theField        String containing name of field for which sample values will be obtained.

### **Returned Value:**

None

### **See Also**

writeFieldSample        sendToServer

writeQueryForm

## **toggleOVMap**

### **Description:**

Toggles the visibility of the overview map display.

Uses: imsURL, ovIsVisible, isIE, and theCursor.

Calls: putExtentOnOVMap, showLayer, and hideLayer in aimsDHTML.js.

Called by: processStartExtent in aimsCommon.js and ToolBar button.

### **Category:**

Extended Map

### **File:**

aimsDHTML.js

### **Syntax:**

toggleOVMap()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

[putExtentOnOVMap](#)

[showLayer](#)

[hideLayer](#)

[processStartExtent](#)

## **untag**

### **Description:**

Replaces tag braces (“<”, “>”) within a string with safe characters (“«”, “»”) so that the string can be displayed on a web page.

Called by: `getBufferAttributeData` in `aimsBuffer.js` and `displayAttributeData` and `parseHyperLink` in `aimsIDentify.js`.

### **Category:**

Basic Map

### **File:**

`aimsCommon.js`

### **Syntax:**

`untag(inputString)`

### **Arguments:**

`inputString`      String to be converted.

### **Returned Value:**

String      Converted string.

### **See Also**

`getBufferAttributeData`      `displayAttributeData`

`parseHyperLink`

## **updateMeasurebox**

### **Description:**

Updates the Measure Box values in Measure mode.

Uses: isNav4, totalMeasure, currentMeasure, and ScaleBarUnits.

Calls: showLayer in aimsDHTML.js.

Called by: resetClick and clickFunction in aimsClick.js, calcDistance in aimsMap.js, clearSelection in aimsSelect.js, and processXML in aimsXML.js.

### **Category:**

Interactive Map

### **File:**

aimsClick.js

### **Syntax:**

updateMeasureBox()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

showLayer

resetClick

clickFunction

calcDistance

clearSelection

processXML

## **useCustomFunction**

### **Description:**

Available for custom processing of returned ArcXML responses. The function processXML will call this function if the value of the global XMLMode is 1,000 or greater.

Uses: XMLMode.

Calls: hideLayer in aimsDHTML.js.

Called by: processXML in aimsXML.js.

### **Category:**

Custom

### **File:**

aimsCustom.js

### **Syntax:**

useCustomFunction(theReply)

### **Arguments:**

theReply      String containing returned ArcXML response.

### **Returned Value:**

None

### **See Also**

hideLayer      processXML

## **writeBlankMapXML**

### **Description:**

Requests a blank map image for the purpose of obtaining adjusted map extent coordinates.

Uses: limitLeft, limitTop, limitRight, limitBottom, iWidth, iHeight, aimsLayersPresent, layerCount, and imsURL.

Uses: LayerName array.

Calls: sendToServer in aimsXML.js.

Called by: processStartExtent in aimsCommon.js.

### **Category:**

Basic Map

### **File:**

aimsXML.js

### **Syntax:**

writeBlankMapXML()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

sendToServer      processStartExtent

## **writeBufferForm**

### **Description:**

Displays a form to set parameters for the Buffer mode.

Uses: useTextFrame, textFrameBackColor, textFrameColor, layerCount, ScaleBarUnits, ActiveLayerIndex, and getBufferData.

Uses: LayerType and LayerName arrays.

Called by: clickFunction in aimsIDentify.js.

### **Category:**

Buffer

### **File:**

aimsBuffer.js

### **Syntax:**

writeBufferForm()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

[clickFunction](#)

## **writeEnvelopeBufferXML**

### **Description:**

Writes ArcXML request string for a buffer using envelope spatial selection.

Uses: swapSelectFields, selectFields, bufferTargetLayerIndex, ActiveLayer, ActiveLayerType, selectEnvelope, useLimitExtent, limitLeft, limitTop, limitRight, limitBottom, bufferDistance, bufferSmoothEdges, and ScaleBarUnits.

Uses: selFieldList and LayerID arrays.

Called by: writeGetBufferData in aimsBuffer.js.

### **Category:**

Buffer

### **File:**

aimsBuffer.js

### **Syntax:**

`writeEnvelopeBufferXML()`

### **Arguments:**

None

### **Returned Value:**

String        ArcXML request string to be sent to QueryServer.

### **See Also**

[writeGetBufferData](#)

# **writeEnvelopeXML**

## **Description:**

Writes an ArcXML request string for an envelope spatial selection to be sent to the QueryServer. The string is not sent to the server by this routine.

Uses: limitLeft, limitTop, limitRight, and limitBottom.

Called by: writeGetFeatures2 and writeGetFeatures3 in aimsSelect.js.

## **Category:**

Select

## **File:**

aimsSelect.js

## **Syntax:**

```
writeEnvelopeXML(theLayer theLayerType, theFields, maxReturned, startRec, theEnvelope,  
hasLimit)
```

## **Arguments:**

theLayer String containing name of layer.

theLayerType String containing layer feature type (point, line, polygon).

theFields String containing list of fields to be returned in response, each separated by a space (ID and shape field required. If all, use “#ALL#”.)

maxReturned Numeric representing number of features to be returned.

startRec Numeric representing record in set of selected features to start count of maxReturned for features to return.

theEnvelope String containing envelope string, consisting of ‘minx=”x.xx” miny=”y.yy” maxx=”x.xx” maxy=”y.yy”’.

hasLimit Boolean indicating if the limit extent should be included in the spatial selection criteria.

## **Returned Value:**

String ArcXML request string to be sent to QueryServer.

## **See Also**

[writeGetFeatures2](#)

[writeGetFeatures3](#)

## **writeFieldSample**

### **Description:**

Writes an ArcXML request for field sample values for the Query form.

Uses: numberDataSamples, ActiveLayer, selectLayer, selectType, ActiveLayerType, selectCount, and highlightedOne.

Uses: selectPoints, selectLeft, selectTop, selectRight, and selectBottom arrays.

Called by: queryForm and tempGetSamples in aimsQuery.js.

### **Category:**

Query

### **File:**

aimsQuery.js

### **Syntax:**

writeFieldSample(theField)

### **Arguments:**

theField      String containing name of field from which to obtain sample values.

### **Returned Value:**

String ArcXML request to be sent to the QueryServer.

### **See Also**

queryForm    tempGetSamples

## **writeFindRequest**

### **Description:**

Writes an ArcXML request for the Find mode.

Uses: maxFeaturesReturned, queryStartRecord, ActiveLayer, useLimitExtent, limitLeft, limitTop, limitRight, limitBottom, selectLayer, selectType, ActiveLayerType, selectCount, and highlightedOne.

Uses: selectPoints, selectLeft, selectTop, selectRight, and selectBottom arrays.

Called by: getFind in aimsQuery.js.

### **Category:**

Query

### **File:**

Query.js

### **Syntax:**

`writeFindRequest(findQuery,fieldList)`

### **Arguments:**

`findQuery`      String containing expression to be used in where clause of a standard SQL query.  
`fieldList`      String containing list of fields to be returned in response, each separated by a space (ID and shape field required. If all, use “#ALL#”.)

### **Returned Value:**

`String`      ArcXML request to be sent to QueryServer.

### **See Also**

`getFind`

## **writeGeocodeXML**

### **Description:**

Writes an ArcXML request to be sent to the GeocodeServer to locate an address.

Uses: GCidCount, maxGeocodeCandidates, minGeocodeScore, and GCActivelayer.

Uses: GCLayerID, GCid, and GCvalue arrays.

Called by: sendQuery function in addmatch.htm.

### **Category:**

Geocode

### **File:**

aimsGeocode.js

### **Syntax:**

`writeGeocodeXML()`

### **Arguments:**

None

### **Returned Value:**

String        ArcXML request string to be sent to GeocodeServer.

### **See Also**

[sendQuery](#)

## **writeGetBufferData**

### **Description:**

Calls appropriate function to write ArcXML request for buffering to send to the QueryServer.

Uses: bufferSmoothEdges, bufferDistance, and selectionMode.

Calls: writeQueryBufferXML, writeEnvelopeBufferXML, and writeShapeBufferXML in aimsBuffer.js.

Called by: processXML in aimsXML.js.

### **Category:**

Buffer

### **File:**

aimsBuffer.js

### **Syntax:**

writeGetBufferData()

### **Arguments:**

None

### **Returned Value:**

String      ArcXML request string to be sent to QueryServer.

### **See Also**

writeQueryBufferXML      writeEnvelopeBufferXML

writeShapeBufferXML      processXML

## **writeGetFeatures**

### **Description:**

Used by IDentify and HyperLink modes to request data.

Uses: swapSelectFields, selectFields, ActiveLayerIndex, ActiveLayer, ActiveLayerType, useLimitExtent, and highlightedOne.

Uses: selectFieldList array.

Calls: writeIDentifyXML in aimsIDentify.js.

Called by: identify and hyperLink in aimsIDentify.js.

### **Category:**

IDentify/Hyperlink

### **File:**

aimsIDentify.js

### **Syntax:**

writeGetFeatures(west1,south1,east1,north1)

### **Arguments:**

west1	Numeric representing west (left) x-coordinate of envelope used in spatial selection.
south1	Numeric representing south (bottom) y-coordinate of envelope used in spatial selection.
east1	Numeric representing east (right) x-coordinate of envelope used in spatial selection.
north1	Numeric representing north (top) y-coordinate of envelope used in spatial selection.

### **Returned Value:**

String ArcXML request string to be sent to the QueryServer.

### **See Also**

[writeIDentifyXML](#)      [identify](#)      [hyperLink](#)

## **writeGetFeatures2**

### **Description:**

Used by Select by Rectangle mode to create ArcXML request for selection.

Uses: swapSelectFields, selectFields, ActiveLayerIndex, ActiveLayer, ActiveLayerType, maxFeaturesReturned, queryStartRecord, useLimitExtent, selectLayer, selectType, selectCount, and highlightedOne.

Uses: selfFieldList, selectPoints, selectLeft, selectTop, selectRight, and selectBottom arrays.

Calls: writeEnvelopeXML in aimsSelect.js.

Called by: select and stopSelectBox in aimsSelect.js.

### **Category:**

Select

### **File:**

aimsSelect.js

### **Syntax:**

writeGetFeatures2(west1,south1,east1,north1)

### **Arguments:**

west1 Numeric representing west (left) x-coordinate of envelope used in spatial selection.

south1 Numeric representing south (bottom) y-coordinate of envelope used in spatial selection.

east1 Numeric representing east (right) x-coordinate of envelope used in spatial selection.

north1 Numeric representing north (top) y-coordinate of envelope used in spatial selection.

### **Returned Value:**

String ArcXML request string to be sent to the QueryServer.

### **See Also**

[writeEnvelopeXML](#)

[select](#)

[stopSelectBox](#)

## **writeGetFeatures3**

### **Description:**

Generates an ArcXML request for envelope spatial selection.

Uses: swapSelectFields, selectFields, ActiveLayerIndex, ActiveLayer, ActiveLayerType, maxFeaturesReturned, queryStartRecord, useLimitExtent, selectLayer, selectType, selectCount, and highlightedOne.

Uses: selFieldList, selectPoints, selectLeft, selectTop, selectRight, and selectBottom arrays.

Calls : writeEnvelopeXML in aimsSelect.js.

Called by: getMoreData in aimsSelect.js.

### **Category:**

Select

### **File:**

aimsSelect.js

### **Syntax:**

writeGetFeatures3()

### **Arguments:**

None

### **Returned Value:**

String        ArcXML request string to be sent to the QueryServer.

### **See Also**

[writeEnvelopeXML](#)    [getMoreData](#)

## **writeIDentifyXML**

### **Description:**

Writes an ArcXML request for the IDentify mode.

Uses: limitLeft, limitTop, limitRight, and limitBottom.

Called by: writeGetFeatures in aimsIDentify.js.

### **Category:**

IDentify/Hyperlink

### **File:**

aimsIDentify.js

### **Syntax:**

`writeIDentifyXML(theLayer, theLayerType, theFields, leftX, bottomY, rightX, topY, hasLimit)`

### **Arguments:**

`theLayer` String containing name of layer.

`theLayerType` String containing layer feature type (point, line, polygon).

`theFields` String containing list of fields to be returned in response, each separated by a space (ID and shape field required. If all, use “#ALL#”.)

`leftX` Numeric representing selection envelope left x-coordinate.

`bottomY` Numeric representing selection envelope bottom y-coordinate.

`rightX` Numeric representing selection envelope right x-coordinate.

`topY` Numeric representing selection envelope top y-coordinate.

`hasLimit` Boolean indicating if the limit extent should be included in the spatial selection criteria.

### **Returned Value:**

String ArcXML request string to be sent to QueryServer.

### **See Also**

[writeGetFeatures](#)

## **writeLayerListForm**

### **Description:**

Displays the LayerList in a separate window. Used when there is no frame called TocFrame.

Loads toc.htm into separate window.

Uses: appDir.

Called by: clickFunction in aimsClick.js and getXYs in aimsXML.js.

### **Category:**

Extended Map

### **File:**

aimsLayers.js

### **Syntax:**

writeLayerListForm()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

[clickFunction](#)      [getXYs](#)

## **writeModeFrame**

### **Description:**

Reloads ModeFrame.htm into the frame called ModeFrame. ModeFrame.htm displays the current viewer mode. The global useModeFrame must be set to True.

Uses: appDir.

Called by: clickFunction in aimsClick.js.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

writeModeFrame(currentMode)

### **Arguments:**

currentMode      String containing name of current viewer mode.

### **Returned Value:**

None

### **See Also**

clickFunction

## **writeModeLayers**

### **Description:**

Updates the Mode display if it is displayed on a style sheet (Netscape layer). The following globals must be set accordingly: useModeFrame=false; drawFloatingMode=true; modeLayerOn=true;

Uses: modeLayerFont, modeLayerShadowColor, modeLayerSize, and modeLayerColor.

Calls: replaceLayerContent in aimsDHTML.js.

Called by: clickFunction in aimsClick.js.

### **Category:**

Basic Map

### **File:**

aimsCommon.js

### **Syntax:**

writeModeLayers(currentMode)

### **Arguments:**

currentMode String containing name of current viewer mode.

### **Returned Value:**

None

### **See Also**

[replaceLayerContent](#)      [clickFunction](#)

## **writeOverlayXML**

### **Description:**

Writes an ArcXML request for a map image displaying only user click points. The image has a transparent background. The global noOverlay must be set to false. This does not work properly in Internet Explorer 4.

Uses: left, top, right, bottom, iHeight, iWidth, layerCount, theServiceName, hostName, theImagePath, theImageURLpath, theImageType, clickCount, clickMarkerColor, clickMarkerType, and clickMarkerSize.

Uses: LayerName, clickPointX, and clickPointY arrays.

Called by: clickAddPoint and deleteClick in aimsClick.js, and processXML in aimsXML.js.

### **Category:**

Interactive Map

### **File:**

aimsClick.js

### **Syntax:**

writeOverlayXML()

### **Arguments:**

None

### **Returned Value:**

String            ArcXML request to be sent to ImageServer.

### **See Also**

clickAddPoint        deleteClick  
processXML

## **writeOVXML**

### **Description:**

Writes an ArcXML request for a map image for the overview map display.

Uses: xDistance, fullWidth, fullHeight, left, bottom, top, right, xHalf, yHalf, fullOVLeft, fullOVTop, fullOVRight, fullOVBottom, i2Width, i2Height, toggleOVVisible, imsURL, imsOVURL, layerCount, mapBackColor, drawOVErcentBox, and ovBoxColor.

Uses: LayerName and LayerVisible array.

Called by: getPrintOV in aimsPrint.js and by processXML in aimsXML.js.

### **Category:**

Basic Map

### **File:**

aimsXML.js

### **Syntax:**

writeOVXML()

### **Arguments:**

None

### **Returned Value:**

String            ArcXML request string to be sent to the ImageServer.

### **See Also**

getPrintOV            processXML

## **writePrintPage**

### **Description:**

Writes a web page that can be sent to the printer containing main map, overview map, and Legend displays. These displays are combined into one page in an HTML table with a user-defined title.

Uses: printTitle, printMapURL, printOVURL, printLegURL, legendVisible, hasOVMMap, and legVis2.

Calls: hideRetrieveMap in aimsMap.js.

Called by: getPrintLegend in aimsPrint.js and processXML in aimsXML.js.

### **Category:**

Print

### **File:**

aimsPrint.js

### **Syntax:**

writePrintPage()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

hideRetrieveMap      getPrintLegend  
processXML

## **writeQueryBufferXML**

### **Description:**

Writes an ArcXML request for a selection within a buffer around features spatially selected by a query. This function does not actually send the string to the server.

Uses: swapSelectFields, selectFields, bufferTargetLayerIndex, ActiveLayer, ActiveLayerType, setQueryString, bufferDistance, bufferSmoothEdges, ScaleBarUnits, useLimitExtent, limitLeft, limitTop, limitRight, and limitBottom.

Uses: selFieldList, LayerType, and LayerID arrays.

Called by: writeGetBufferData in aimsBuffer.js.

### **Category:**

Buffer

### **File:**

aimsBuffer.js

### **Syntax:**

`writeQueryBufferXML()`

### **Arguments:**

None

### **Returned Value:**

String        ArcXML request string to be sent to the QueryServer.

### **See Also**

[writeGetBufferData](#)

## **writeQueryForm**

### **Description:**

Displays a Query form for attribute queries.

Uses: fieldIndex, showSampleValues, dQuote, useTextFrame, textFrameBackColor, and LayerFieldCount.

Uses: LayerFields, LayerFieldType, and selectData arrays.

Called by: queryForm in aimsQuery.js and the form created by this function.

### **Category:**

Query

### **File:**

aimsQuery.js

### **Syntax:**

writeQueryForm()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

queryForm

## **writeQueryXML**

### **Description:**

Writes an ArcXML request for an attribute query.

Uses: swapSelectFields, selectFields, ActiveLayerIndex, maxFeaturesReturned, queryStartRecord, ActiveLayer, useLimitExtent, limitLeft, limitTop, limitRight, limitBottom, selectLayer, selectType, ActiveLayerType, selectCount, and highlightedOne.

Uses: selFieldList, selectPoints, selectLeft, selectRight, selectTop, and selectBottom arrays.

Called by: sendQueryString, sendStoredQuery, and getMoreData in aimsQuery.js.

### **Category:**

Query

### **File:**

aimsQuery.js

### **Syntax:**

`writeQueryXML(queryString)`

### **Arguments:**

queryString      String containing expression to be used in where clause of a standard SQL query.

### **Returned Value:**

String      ArcXML request string to be sent to the QueryServer.

### **See Also**

`sendQueryString`      `sendStoredQuery`

`getMoreData`

## **writeShapeBufferXML**

### **Description:**

Function that writes an ArcXML request for a selection within a buffer around features spatially selected by a shape. Shapes are lines or polygons.

Uses: swapSelectFields, selectFields, bufferTargetLayerIndex, ActiveLayer, ActiveLayerType, useLimitExtent, limitRight, limitTop, limitLeft, limitBottom, clickCount, bufferDistance, bufferSmoothEdges, and ScaleBarUnits.

Uses: selFieldList, clickPointX, clickPointY, and LayerID arrays.

Called by: writeGetBufferData in aimsBuffer.js.

### **Category:**

Buffer

### **File:**

aimsBuffer.js

### **Syntax:**

writeShapeBufferXML(theType)

### **Arguments:**

theType      Numeric representing type of shape used in selecting features to be buffered  
(1=line, 2=polygon).

### **Returned Value:**

String      ArcXML request string to be sent to the QueryServer.

### **See Also**

[writeGetBufferData](#)

## **writeShapeSelect**

### **Description:**

Writes an ArcXML request for a selection of features using a shape. Shapes are lines or polygons.

Uses: swapSelectFields, selectFields, ActiveLayerIndex, maxFeaturesReturned, queryStartRecord, ActiveLayer, ActiveLayerType, useLimitExtent, limitRight, limitTop, limitLeft, limitBottom, clickCount, bufferDistance, bufferSmoothEdges, ScaleBarUnits, selectLayer, selectLayerType, selectCount, and highlightedOne.

Uses: selFieldList, clickPointX, clickPointY, selectPoint, selectLeft, selectRight, selectTop, selectBottom arrays.

Called by: sendShapeSelect and getMoreData in aimsSelect.js.

### **Category:**

Select

### **File:**

aimsSelect.js

### **Syntax:**

writeShapeSelect(theType)

### **Arguments:**

theType      Numeric representing type of shape used in selecting features to be buffered  
(1=line, 2=polygon).

### **Returned Value:**

String      ArcXML request string to be sent to the QueryServer.

### **See Also**

sendShapeSelect      getMoreData

## **writeStoredQueryForm**

### **Description:**

Displays a Query form for stored Queries.

Uses: storedQueryIndex, useTextFrame, textFrameFormColor, ActiveLayerIndex, and storedQueryCount.

Uses: LayerName, storedQueryName and storedQueryVarCount arrays.

Calls: sendStoredQuery in aimsQuery.js.

Called by: storedQueryForm in aimsQuery.js and the form this function creates.

### **Category:**

Query

### **File:**

aimsQuery.js

### **Syntax:**

`writeStoredQueryForm(theIndex)`

### **Arguments:**

`theIndex`      String containing returned ArcXML response.

### **Returned Value:**

None

### **See Also**

`storedQueryForm`      `sendStoredQuery`

## **writeXML**

### **Description:**

Writes an ArcXML request for a new map image. This is the default request sent for a main map display image.

Uses: left, top, right, bottom, iHeight, iWidth, aimsLayersPresent, toggleVisible, layerCount, aimsClassRenderPresent, mapBackColor, aimsLegendPresent, legendVisible, aimsBufferPresent, showBuffer, aimsSelectPresent, showGeocode, geocodeX, geocodeY, geocodePointColor, geocodePointSize, geocodeLabel, aimsClickPresent, noOverlay, clickCount, selectColor, clickType, clickMarkerColor, clickMarkerType, clickMarkerSize, aimsCustomPresent, drawCopyright, CopyrightCoords, CopyrightText, CopyrightStyle, CopyrightSize, CopyrightFont, CopyrightColor, CopyrightBackground, CopyrightBColor, CopyrightGlow, CopyrightGlowColor, drawNorthArrow, NorthArrowType, NorthArrowSize, NorthArrowCoords, NorthArrowAngle, drawScaleBar, ScaleBarbackColor, ScaleBarFont, ScaleBarStyle, ScaleBarColor, MapUnits, scalebarFontColor, ScaleBarUnits, ScaleBarPrecision, ScaleBarSize, ScaleBarWidth, drawModeOnMap, modeBlurb, modeMapColor, and modeMapGlow.

Uses: LayerVisible, LayerID, clickPointX, and clickPointY arrays.

Calls: addSpecialRenderToMap in aimsClassRender.js (custom sample), addBufferToMap in aimsBuffer.js, addLegendToMap in aimsLegend.js, addCustomToMap1, addCustomToMap2, addCustomToMap3, and addCustomToMap4 in aimsCustom.js, addSelectToMap in aimsSelect.js, and getScaleBarDistance in aimsMap.js.

Called by: clickAddPoint, resetClick, and deleteClick in aimsClick.js, by getLegend in aimsLegend.js, getPrintMap in aimsPrint.js, clearSelection in aimsSelect.js, and sendMapXML and justGetMap in aimsXML.js.

### **Category:**

Basic Map

### **File:**

aimsXML.js

### **Syntax:**

writeXML()

## **writeXML (continued)**

### **Arguments:**

None

### **Returned Value:**

String      ArcXML request string to be sent to the ImageServer.

### **See Also**

addSelectToMap	addCustomToMap1	addBufferToMap	addCustomToMap2
addLegendToMap	addCustomToMap3	addScaleBarDistance	addCustomToMap4
clickAddPoint	getPrintMap	resetClick	clearSelection
deleteClick	sendMapXML	getLegend	justGetMap

## **zoomBack**

### **Description:**

Sets the map extent to the previous map extent and requests a new map image.

Uses: aimsDHTMLPresent, left, right, top, bottom, lastLeft, lastRight, lastTop, hspc, vspc, and lastBottom.

Calls: moveLayer in aimsDHTML.js and sendMapXML in aimsXML.js.

Called by: clickFunction in aimsClick.js.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

zoomBack()

### **Arguments:**

None

### **Returned Value:**

None

### **See Also**

moveLayer                  sendMapXML

clickFunction

## **zoomButton**

### **Description:**

Zooms the map extent in or out. Zoom factor is either double or half the current extent.

Uses: left, right, top, bottom, xHalf, and yHalf.

Calls: saveLastExtent and checkFullExtent in aimsMap.js and sendMapXML in aimsXML.js.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

zoomButton(zoomType)

### **Arguments:**

zoomType      Numeric representing type of zoom (1=in, any other number=out).

### **Returned Value:**

None

### **See Also**

saveLastExtent      checkFullExtent

sendMapXML

## **zoomin**

### **Description:**

Zooms in to the area surrounding the click point on the main map display. Used as an alternative to the interactive ZoomBox mode. Zoom factor is determined by the global zoomFactor.

Uses: mouseX, mouseY, lastLeft, lastRight, lastTop, lastBottom, mapX, mapY, left, right, top, bottom, xHalf, yHalf, and zoomFactor.

Calls: getMapXY in aimsNavigation.js, saveLastExtent in aimsMap.js, and sendMapXML in aimsXML.js.

Called by: stopZoomBox in aimsNavigation.js.

### **Category:**

Interactive Map

### **File:**

aimsNavigation.js

### **Syntax:**

zoomin(e)

### **Arguments:**

e                  Event from browser.

### **Returned Value:**

None

### **See Also**

stopZoomBox	saveLastExtent
sendMapXML	getMapXY

## **zoomout**

### **Description:**

Zooms out from the area surrounding the click point on the main map display. Used as an alternative to the interactive ZoomBox mode. Zoom factor is determined by the global zoomFactor.

Uses: mouseX, mouseY, lastLeft, lastRight, lastTop, lastBottom, mapX, mapY, xDistance, yDistance, left, right, top, bottom, and zoomFactor.

Calls: getMapXY in aimsNavigation.js, saveLastExtent in aimsMap.js, and sendMapXML in aimsXML.js.

Called by: stopZoomOutBox in aimsNavigation.js.

### **Category:**

Interactive Map

### **File:**

aimsNavigation.js

### **Syntax:**

zoomout(e)

### **Arguments:**

e              Event from browser.

### **Returned Value:**

None

### **See Also**

stopZoomBox

getMapXY

saveLastExtent

sendMapXML

## **zoomScale**

### **Description:**

Sets main map display extent to a factor of the full extent, where 1.0 is 100 percent of full extent. The display is centered on the center of the full extent.

Uses: fullWidth, fullHeight, xDistance, yDistance, left, top, right, and bottom.

Calls: saveLastExtent in aimsMap.js and sendMapXML in aimsXML.js.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

zoomScale(inScale)

### **Arguments:**

inScale Numeric representing scale factor between 0.0 and 1.0, where 1.0 is full extent.

### **Returned Value:**

None

### **See Also**

[saveLastExtent](#)

[sendMapXML](#)

## **zoomToEnvelope**

### **Description:**

Zooms to the envelope defined by coordinates passed in the arguments.

Uses: left, bottom, right, and top.

Calls: saveLastExtent in aimsMap.js and sendMapXML in aimsXML.js.

Called by: clickFunction in aimsClick.js.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

`zoomToEnvelope(minXin,minYin,maxXin,maxYin)`

### **Arguments:**

<code>minXin</code>	Numeric representing envelope left x-coordinate.
<code>minYin</code>	Numeric representing envelope bottom y-coordinate.
<code>maxXin</code>	Numeric representing envelope right x-coordinate.
<code>maxYin</code>	Numeric representing envelope top y-coordinate.

### **Returned Value:**

None

### **See Also**

`saveLastExtent`      `sendMapXML`  
`clickFunction`

## **zoomToPoint**

### **Description:**

Zooms to area around the point defined by the coordinates passed in the arguments.

Uses: limitLeft, limitRight, limitTop, limitBottom, selectPointMargin, left, right, top, bottom, showGeocode, geocodeX, geocodeY, and geocodeLabel.

Calls: saveLastExtent in aimsMap.js and sendMapXML in aimsXML.js.

Called by: link in table display created by parseGeocodeResults.

### **Category:**

Basic Map

### **File:**

aimsMap.js

### **Syntax:**

zoomToPoint(xIn, yIn, drawIt, theLabel)

### **Arguments:**

xIn            Numeric representing map x-coordinate of point.

yIn            Numeric representing map y-coordinate of point.

drawIt        Boolean indicating if the point should be drawn on map.

theLabel      String containing label to be displayed by point. If no label is to be written, an empty string ("") is passed.

### **Returned Value:**

None

### **See Also**

saveLastExtent      parseGeocodeResults

sendMapXML

## Global variables by name (ActiveLayer–chkUnits)

Variable Name	File	Category
ActiveLayer	aimsLayers.js	Extended Map
ActiveLayerIndex	ArclMSparam.js	Basic Map
ActiveLayerType	aimsLayers.js	Extended Map
aimsBufferPresent	MapFrame.htm	Basic Map
aimsClassRenderPresent	MapFrame.htm	Basic Map
aimsClickPresent	MapFrame.htm	Basic Map
aimsCommonPresent	MapFrame.htm	Basic Map
aimsCustomPresent	MapFrame.htm	Basic Map
aimsDHTMLPresent	MapFrame.htm	Basic Map
aimsGenericPresent	MapFrame.htm	Basic Map
aimsGeocodePresent	MapFrame.htm	Basic Map
aimsIDIdentifyPresent	MapFrame.htm	Basic Map
aimsLayersPresent	MapFrame.htm	Basic Map
aimsLegendPresent	MapFrame.htm	Basic Map
aimsMapPresent	MapFrame.htm	Basic Map
aimsNavigationPresent	MapFrame.htm	Basic Map
aimsOptionsPresent	MapFrame.htm	Basic Map
aimsPrint Present	MapFrame.htm	Basic Map
aimsQueryPresent	MapFrame.htm	Basic Map
aimsSelectPresent	MapFrame.htm	Basic Map
aimsXMLPresent	MapFrame.htm	Basic Map
AliasFieldNames	aimsLayers.js	Extended Map
AliasFieldAliases	aimsLayers.js	Extended Map
allowOptions	ArclMSparam.js	Basic Map
appDir	aimsMap.js	Basic Map
blankImage	aimsClick.js	Interactive
bottom	aimsMap.js	Basic Map
bufferDistance	aimsBuffer.js	Buffer
bufferSmoothEdges	aimsBuffer.js	Buffer
bufferTargetLayer	aimsBuffer.js	Buffer
bufferTargetLayerIndex	aimsBuffer.js	Buffer
canLoad	aimsMap.js	Basic Map
canQuery	aimsIDIdentify.js	IDIdentify/Hyperlink
catURL	ArclMSparam.js	Basic Map
chkUnits	aimsCommon.js	Basic Map

## Global variables by name (ClassRendererLayer–drawTargetLayer)

Variable Name	File	Category
ClassRenderLayer	ArclMSparam.js	Basic Map
ClassRenderString	ArclMSparam.js	Basic Map
clickCount	aimsClick.js	Interactive
clickMarkerColor	ArclMSparam.js	Basic Map
clickMarkerSize	ArclMSparam.js	Basic Map
clickMarkerType	ArclMSparam.js	Basic Map
clickMeasure	aimsClick.js	Interactive
clickPointX	aimsClick.js	Interactive
clickPointY	aimsClick.js	Interactive
clickType	aimsClick.js	Interactive
CopyrightBackground	ArclMSparam.js	Basic Map
CopyrightBGColor	ArclMSparam.js	Basic Map
CopyrightColor	ArclMSparam.js	Basic Map
CopyrightCoords	ArclMSparam.js	Basic Map
CopyrightFont	ArclMSparam.js	Basic Map
CopyrightGlow	ArclMSparam.js	Basic Map
CopyrightGlowColor	ArclMSparam.js	Basic Map
CopyrightSize	ArclMSparam.js	Basic Map
CopyrightStyle	ArclMSparam.js	Basic Map
CopyrightText	ArclMSparam.js	Basic Map
currentHyperLinkField	aimsIDeIdentify.js	IDeIdentify/Hyperlink
currentHyperLinkLayer	aimsIDeIdentify.js	IDeIdentify/Hyperlink
currentMeasure	aimsClick.js	Interactive
debugOn	aimsMap.js	Basic Map
defaultLegTitle	aimsLegend.js	Legend
displayLayerInfoButton	aimsLayers.js	Extended Map
dQuote	aimsMap.js	Basic Map
drawCopyright	ArclMSparam.js	Basic Map
drawFloatingMode	ArclMSparam.js	Basic Map
drawLegendOnly	aimsLegend.js	Legend
drawModeOnMap	ArclMSparam.js	Basic Map
drawNorthArrow	ArclMSparam.js	Basic Map
drawOVExtentBox	aimsXML.js	Basic Map
drawScaleBar	ArclMSparam.js	Basic Map
drawSelectBoundary	aimsCommon.js	Basic Map
drawTargetLayer	aimsBuffer.js	Buffer

## Global variables by name (enforceFullExtent–geocodeLabelSize)

Variable Name	File	Category
enforceFullExtent	aimsMap.js	Basic Map
FeatureLayerCount	aimsLayers.js	Extended Map
fieldAliasList	ArclMSparam.js	Basic Map
fieldIndex	aimsLayers.js	Extended Map
findXMLMode	aimsXML.js	Basic Map
footerFilePath	ArclMSparam.js	Basic Map
formFilePath	ArclMSparam.js	Basic Map
fullBottom	aimsMap.js	Basic Map
fullHeight	aimsMap.js	Basic Map
fullLeft	aimsMap.js	Basic Map
fullOVBottom	aimsMap.js	Basic Map
fullOVHeight	aimsMap.js	Basic Map
fullOVLeft	aimsMap.js	Basic Map
fullOVRRight	aimsMap.js	Basic Map
fullOVTop	aimsMap.js	Basic Map
fullOVWidth	aimsMap.js	Basic Map
fullRight	aimsMap.js	Basic Map
fullTop	aimsMap.js	Basic Map
fullWidth	aimsMap.js	Basic Map
GCActiveLayer	aimsGeocode.js	Geocode
GCaddress	aimsGeocode.js	Geocode
GCdesc	aimsGeocode.js	Geocode
GCid	aimsGeocode.js	Geocode
GCidCount	aimsGeocode.js	Geocode
GClabel	aimsGeocode.js	Geocode
GCLayerCount	aimsGeocode.js	Geocode
GCLayerID	aimsGeocode.js	Geocode
GCLayers	aimsGeocode.js	Geocode
GCLayerStyle	aimsGeocode.js	Geocode
GCpointCount	aimsGeocode.js	Geocode
GCpointX	aimsGeocode.js	Geocode
GCpointY	aimsGeocode.js	Geocode
GCscore	aimsGeocode.js	Geocode
GCvalue	aimsGeocode.js	Geocode
geocodeLabel	aimsMap.js	Basic Map
geocodeLabelSize	ArclMSparam.js	Basic Map

## Global variables by name (geocodePointColor–isNav6up)

Variable Name	File	Category
geocodePointColor	ArcIMSParam.js	Basic Map
geocodePointSize	ArcIMSParam.js	Basic Map
geocodeX	aimsMap.js	Basic Map
geocodeY	aimsMap.js	Basic Map
getBufferData	aimsBuffer.js	Buffer
getLimitExtent	aimsMap.js	Basic Map
getStartingExtent	aimsMap.js	Basic Map
hasOVMap	ArcIMSParam.js	Basic Map
hasTOC	ArcIMSParam.js	Basic Map
hasToolBarOnLayer	ArcIMSParam.js	Basic Map
headerFilePath	ArcIMSParam.js	Basic Map
highlightColor	ArcIMSParam.js	Basic Map
highlightedOne	aimsSelect.js	Select
hostName	ArcIMSParam.js	Basic Map
hspc	ArcIMSParam.js	Basic Map
hyperLinkFields	ArcIMSParam.js	Basic Map
hyperLinkLayers	ArcIMSParam.js	Basic Map
hyperlinkWindowHeight	aimsIDeIdentify.js	IDeIdentify/Hyperlink
hyperlinkWindowWidth	aimsIDeIdentify.js	IDeIdentify/Hyperlink
hyperlinkXMLMode	aimsXML.js	Basic Map
i2Height	aimsMap.js	Basic Map
i2Width	aimsMap.js	Basic Map
identifyXMLMode	aimsXML.js	Basic Map
iHeight	aimsMap.js	Basic Map
imageLimitBottom	aimsMap.js	Basic Map
imageLimitLeft	aimsMap.js	Basic Map
imageLimitRight	aimsMap.js	Basic Map
imageLimitTop	aimsMap.js	Basic Map
imsGeocodeURL	ArcIMSParam.js	Basic Map
imsOVURL	ArcIMSParam.js	Basic Map
imsQueryURL	ArcIMSParam.js	Basic Map
imsURL	ArcIMSParam.js	Basic Map
isIE	ArcIMSParam.js	Basic Map
isNav	ArcIMSParam.js	Basic Map
isNav4	ArcIMSParam.js	Basic Map
isNav5up	ArcIMSParam.js	Basic Map

## Global variables by name (iWidth–legendVisible)

Variable Name	File	Category
iWidth	aimsMap.js	Basic Map
lastBottom	aimsMap.js	Basic Map
lastLeft	aimsMap.js	Basic Map
lastRight	aimsMap.js	Basic Map
lastTop	aimsMap.js	Basic Map
lastTotMeasure	aimsClick.js	Interactive
LayerBottom	aimsLayers.js	Extended Map
layerCount	aimsLayers.js	Extended Map
LayerExtent	aimsLayers.js	Extended Map
LayerFieldCount	aimsLayers.js	Extended Map
LayerFieldList	aimsLayers.js	Extended Map
LayerFieldPrecisionList	aimsLayers.js	Extended Map
LayerFields	aimsLayers.js	Extended Map
LayerFieldSizeList	aimsLayers.js	Extended Map
LayerFieldType	aimsLayers.js	Extended Map
LayerFieldTypeList	aimsLayers.js	Extended Map
LayerID	aimsLayers.js	Extended Map
LayerIDField	aimsLayers.js	Extended Map
LayerIsFeature	aimsLayers.js	Extended Map
LayerLeft	aimsLayers.js	Extended Map
LayerListOpen	aimsLayers.js	Extended Map
LayerMaxScale	aimsLayers.js	Extended Map
LayerMinScale	aimsLayers.js	Extended Map
LayerName	aimsLayers.js	Extended Map
LayerRenderString	aimsLayers.js	Extended Map
LayerRight	aimsLayers.js	Extended Map
LayerShapeField	aimsLayers.js	Extended Map
LayerTop	aimsLayers.js	Extended Map
LayerType	aimsLayers.js	Extended Map
LayerVisible	aimsLayers.js	Extended Map
left	aimsMap.js	Basic Map
leftButton	aimsClick.js	Interactive
legendImage	aimsCommon.js	Basic Map
legendTemp	aimsCommon.js	Basic Map
legendVisible	aimsMap.js	Basic Map

## Global variables by name (legFont–NorthArrowType)

Variable Name	File	Category
legFont	ArcIMSparam.js	Basic Map
legHeight	ArcIMSparam.js	Basic Map
legTitle	ArcIMSparam.js	Basic Map
legVis2	aimsPrint.js	Print
legWidth	ArcIMSparam.js	Basic Map
limitBottom	ArcIMSparam.js	Basic Map
limitLeft	ArcIMSparam.js	Basic Map
limitRight	ArcIMSparam.js	Basic Map
limitTop	ArcIMSparam.js	Basic Map
listAllLayers	ArcIMSparam.js	Basic Map
mapBackColor	ArcIMSparam.js	Basic Map
mapScaleFactor	aimsMap.js	Basic Map
MapUnits	ArcIMSparam.js	Basic Map
mapX	aimsMap.js	Basic Map
mapY	aimsMap.js	Basic Map
maxFeaturesReturned	ArcIMSparam.js	Basic Map
maxGeocodeCandidates	ArcIMSparam.js	Basic Map
minGeocodeScore	ArcIMSparam.js	Basic Map
modeBlurb	aimsCommon.js	Basic Map
modeLayerColor	ArcIMSparam.js	Basic Map
modeLayerFont	ArcIMSparam.js	Basic Map
modeLayerOn	ArcIMSparam.js	Basic Map
modeLayerShadowColor	ArcIMSparam.js	Basic Map
modeLayerSize	ArcIMSparam.js	Basic Map
modeMapColor	ArcIMSparam.js	Basic Map
modeMapGlow	ArcIMSparam.js	Basic Map
modeRefreshMap	ArcIMSparam.js	Basic Map
mouseX	aimsClick.js	Interactive
mouseY	aimsClick.js	Interactive
newSelectCount	aimsIDentify.js	IDeIdentify/Hyperlink
noOverlay	aimsCommon.js	Basic Map
NorthArrowAngle	ArcIMSparam.js	Basic Map
NorthArrowCoords	ArcIMSparam.js	Basic Map
NorthArrowSize	ArcIMSparam.js	Basic Map
NorthArrowType	ArcIMSparam.js	Basic Map

## Global variables by name (number-ScaleBarColor)

Variable Name	File	Category
numberDataSamples	ArcIMSparam.js	Basic Map
numDecimals	ArcIMSparam.js	Basic Map
numStatDecimals	ArcIMSParams.js	Basic Map
okToSend	aimsXML.js	Basic Map
ovBorderWidth	ArcIMSparam.js	Basic Map
ovBoxColor	ArcIMSparam.js	Basic Map
ovBoxSize	ArcIMSparam.js	Basic Map
ovExtentBoxSize	ArcIMSparam.js	Basic Map
ovHspc	ArcIMSparam.js	Basic Map
ovImageVar	ArcIMSparam.js	Basic Map
ovIsVisible	aimsCommon.js	Basic Map
ovMapIsLayer	ArcIMSparam.js	Basic Map
ovVspc	ArcIMSparam.js	Basic Map
panFactor	ArcIMSparam.js	Basic Map
panning	aimsClick.js	Interactive
panX	aimsMap.js	Basic Map
panY	aimsMap.js	Basic Map
pastStart	aimsXML.js	Basic Map
pixelTolerance	ArcIMSparam.js	Basic Map
pixelX	aimsMap.js	Basic Map
pixelY	aimsMap.js	Basic Map
printLegURL	aimsPrint.js	Print
printMapURL	aimsPrint.js	Print
printOVURL	aimsPrint.js	Print
printTitle	aimsPrint.js	Print
queryMode	aimsSelect.js	Select
queryOpen	aimsLayers.js	Extended Map
queryStartRecord	aimsIDeIdentify.js	IDeIdentify/Hyperlink
queryTool	aimsCommon.js	Basic Map
queryXMLMode	aimsXML.js	Basic Map
right	aimsMap.js	Basic Map
rightButton	aimsClick.js	Interactive
ScaleBarBackColor	ArcIMSparam.js	Basic Map
ScaleBarBackground	ArcIMSparam.js	Basic Map
ScaleBarColor	ArcIMSparam.js	Basic Map

## Global variables by name (ScaleBarFont–ShowSampleValues)

Variable Name	File	Category
ScaleBarFont	ArcIMSParam.js	Basic Map
ScaleBarFontColor	ArcIMSParam.js	Basic Map
ScaleBarPrecision	ArcIMSParam.js	Basic Map
ScaleBarRound	ArcIMSParam.js	Basic Map
ScaleBarSize	ArcIMSParam.js	Basic Map
ScaleBarStyle	ArcIMSParam.js	Basic Map
ScaleBarUnits	ArcIMSParam.js	Basic Map
ScaleBarWidth	ArcIMSParam.js	Basic Map
searchTolerance	aimsIDeIdentify.js	IDeIdentify/Hyperlink
selectBlurb	aimsSelect.js	Select
selectBottom	aimsSelect.js	Select
selectBox	aimsClick.js	Interactive
selectColor	ArcIMSParam.js	Basic Map
selectCount	aimsSelect.js	Select
selectData	aimsSelect.js	Select
selectEnvelope	aimsSelect.js	Select
selectFields	ArcIMSParam.js	Basic Map
selectionMode	aimsSelect.js	Select
selectLayer	aimsSelect.js	Select
selectLeft	aimsSelect.js	Select
selectMargin	ArcIMSParam.js	Basic Map
selectPointMargin	ArcIMSParam.js	Basic Map
selectPoints	aimsIDeIdentify.js	IDeIdentify/Hyperlink
selectRight	aimsSelect.js	Select
selectTop	aimsSelect.js	Select
selectType	aimsSelect.js	Select
selectXMLMode	aimsXML.js	Basic Map
selFieldList	ArcIMSParam.js	Basic Map
serverURL	ArcIMSParam.js	Basic Map
setDebug	aimsMap.js	Basic Map
setMapUnits	ArcIMSParam.js	Basic Map
setQueryString	aimsSelect.js	Select
showBuffer	aimsCommon.js	Basic Map
showGeocode	aimsMap.js	Basic Map
showSampleValues	aimsQuery.js	Query

## Global variables by name (showScalePercent=top)

Variable Name	File	Category
showScalePercent	ArcIMSparam.js	Basic Map
showSelectedData	ArcIMSparam.js	Basic Map
showSelectedFeatures	ArcIMSparam.js	Basic Map
showTOC	ArcIMSparam.js	Basic Map
showXYs	ArcIMSparam.js	Basic Map
sQuote	aimsMap.js	Basic Map
startBottom	ArcIMSparam.js	Basic Map
startLeft	ArcIMSparam.js	Basic Map
startRight	ArcIMSparam.js	Basic Map
startTop	ArcIMSparam.js	Basic Map
storedQueryCount	aimsQuery.js	Query
storedQueryFieldList	aimsQuery.js	Query
storedQueryIndex	aimsQuery.js	Query
storedQueryName	aimsQuery.js	Query
storedQueryString	aimsQuery.js	Query
storedQueryVarCount	aimsQuery.js	Query
storedQueryVariable	aimsQuery.js	Query
swapSelectFields	ArcIMSparam.js	Basic Map
textBackColor	ArcIMSparam.js	Basic Map
textFrameBackColor	ArcIMSparam.js	Basic Map
textFrameFormColor	ArcIMSparam.js	Basic Map
textFrameLinkColor	ArcIMSparam.js	Basic Map
textFrameTextColor	ArcIMSparam.js	Basic Map
theBrowser	aimsCommon.js	Basic Map
theCursor	aimsMap.js	Basic Map
theImagePath	aimsCommon.js	Basic Map
theImageType	aimsXML.js	Basic Map
theImageURLPath	aimsCommon.js	Basic Map
theServiceName	aimsCommon.js	Basic Map
theVersion	aimsCommon.js	Basic Map
theVersNum	aimsCommon.js	Basic Map
toggleOVVisible	ArcIMSparam.js	Basic Map
toggleVisible	ArcIMSparam.js	Basic Map
toolMode	aimsMap.js	Basic Map
top	aimsMap.js	Basic Map

## Global variables by name (totalMeasure–x1)

Variable Name	File	Category
totalMeasure	aimsClick.js	Interactive
transparentLevel	ArcIMSpParam.js	Basic Map
useBuffer	ArcIMSpParam.js	Basic Map
useBufferShape	ArcIMSpParam.js	Basic Map
useClearSelect	ArcIMSpParam.js	Basic Map
useExternalWindow	ArcIMSpParam.js	Basic Map
useExtract	ArcIMSpParam.js	Basic Map
useFieldAlias	ArcIMSpParam.js	Basic Map
useFind	ArcIMSpParam.js	Basic Map
useFullExtent	ArcIMSpParam.js	Basic Map
useGeocode	ArcIMSpParam.js	Basic Map
useHyperLink	ArcIMSpParam.js	Basic Map
useIdentify	ArcIMSpParam.js	Basic Map
useLimitExtent	aimsMap.js	Basic Map
useMeasure	ArcIMSpParam.js	Basic Map
useModeFrame	ArcIMSpParam.js	Basic Map
usePan	ArcIMSpParam.js	Basic Map
usePanEast	ArcIMSpParam.js	Basic Map
usePanNorth	ArcIMSpParam.js	Basic Map
usePanSouth	ArcIMSpParam.js	Basic Map
usePanWest	ArcIMSpParam.js	Basic Map
usePrint	ArcIMSpParam.js	Basic Map
useQuery	ArcIMSpParam.js	Basic Map
useReverseGeocode	ArcIMSpParam.js	Basic Map
useSelect	ArcIMSpParam.js	Basic Map
useSetUnits	ArcIMSpParam.js	Basic Map
useStoredQuery	ArcIMSpParam.js	Basic Map
useTextFrame	ArcIMSpParam.js	Basic Map
useZoomActive	ArcIMSpParam.js	Basic Map
useZoomIn	ArcIMSpParam.js	Basic Map
useZoomLast	ArcIMSpParam.js	Basic Map
useZoomOut	ArcIMSpParam.js	Basic Map
vspc	ArcIMSpParam.js	Basic Map
webParams	ArcIMSpParam.js	Basic Map
x1	aimsClick.js	Interactive

## Global variables by name (x2–ztop)

Variable Name	File	Category
x2	aimsClick.js	Interactive
xDistance	aimsMap.js	Basic Map
xHalf	aimsXML.js	Basic Map
xmlEndPos	aimsXML.js	Basic Map
XMLMode	aimsXML.js	Basic Map
y1	aimsClick.js	Interactive
y2	aimsClick.js	Interactive
yDistance	aimsMap.js	Basic Map
yHalf	aimsXML.js	Basic Map
zbottom	aimsClick.js	Interactive
zleft	aimsClick.js	Interactive
zoomBoxColor	ArclMSparam.js	Basic Map
zoomFactor	ArclMSparam.js	Basic Map
zooming	aimsClick.js	Interactive
zoomToSingleSelect	ArclMSparam.js	Basic Map
zright	aimsClick.js	Interactive
ztop	aimsClick.js	Interactive

## Global variables by Category (Basic Map)

<b>Category</b>	<b>File</b>	<b>Variable Name</b>
Basic Map	aimsCommon.js	chkUnits
Basic Map	aimsCommon.js	drawSelectBoundary
Basic Map	aimsCommon.js	legendImage
Basic Map	aimsCommon.js	legendTemp
Basic Map	aimsCommon.js	modeBlurb
Basic Map	aimsCommon.js	noOverlay
Basic Map	aimsCommon.js	ovIsVisible
Basic Map	aimsCommon.js	queryTool
Basic Map	aimsCommon.js	showBuffer
Basic Map	aimsCommon.js	theBrowser
Basic Map	aimsCommon.js	theImagePath
Basic Map	aimsCommon.js	theImageURLPath
Basic Map	aimsCommon.js	theServiceName
Basic Map	aimsCommon.js	theVersion
Basic Map	aimsCommon.js	theVersNum
Basic Map	aimsMap.js	appDir
Basic Map	aimsMap.js	bottom
Basic Map	aimsMap.js	canLoad
Basic Map	aimsMap.js	debugOn
Basic Map	aimsMap.js	dQuote
Basic Map	aimsMap.js	enforceFullExtent
Basic Map	aimsMap.js	fullBottom
Basic Map	aimsMap.js	fullHeight
Basic Map	aimsMap.js	fullLeft
Basic Map	aimsMap.js	fullOVBottom
Basic Map	aimsMap.js	fullOVHeight
Basic Map	aimsMap.js	fullOVLeft
Basic Map	aimsMap.js	fullOVRRight
Basic Map	aimsMap.js	fullOVTop
Basic Map	aimsMap.js	fullOVWidth
Basic Map	aimsMap.js	fullRight
Basic Map	aimsMap.js	fullTop
Basic Map	aimsMap.js	fullWidth
Basic Map	aimsMap.js	geocodeLabel
Basic Map	aimsMap.js	geocodeX

## Global variables by Category (Basic Map continued)

Category	File	Variable Name
Basic Map	aimsMap.js	geocodeY
Basic Map	aimsMap.js	getLimitExtent
Basic Map	aimsMap.js	getStartingExtent
Basic Map	aimsMap.js	i2Height
Basic Map	aimsMap.js	i2Width
Basic Map	aimsMap.js	iHeight
Basic Map	aimsMap.js	imageLimitBottom
Basic Map	aimsMap.js	imageLimitLeft
Basic Map	aimsMap.js	imageLimitRight
Basic Map	aimsMap.js	imageLimitTop
Basic Map	aimsMap.js	iWidth
Basic Map	aimsMap.js	lastBottom
Basic Map	aimsMap.js	lastLeft
Basic Map	aimsMap.js	lastRight
Basic Map	aimsMap.js	lastTop
Basic Map	aimsMap.js	left
Basic Map	aimsMap.js	legendVisible
Basic Map	aimsMap.js	mapScaleFactor
Basic Map	aimsMap.js	mapX
Basic Map	aimsMap.js	mapY
Basic Map	aimsMap.js	panX
Basic Map	aimsMap.js	panY
Basic Map	aimsMap.js	pixelX
Basic Map	aimsMap.js	pixelY
Basic Map	aimsMap.js	right
Basic Map	aimsMap.js	setDebug
Basic Map	aimsMap.js	showGeocode
Basic Map	aimsMap.js	sQuote
Basic Map	aimsMap.js	theCursor
Basic Map	aimsMap.js	toolMode
Basic Map	aimsMap.js	top
Basic Map	aimsMap.js	useLimitExtent
Basic Map	aimsMap.js	xDistance
Basic Map	aimsMap.js	yDistance

## Global variables by Category (Basic Map continued)

<b>Category</b>	<b>File</b>	<b>Variable Name</b>
Basic Map	aimsXML.js	drawOVExtentBox
Basic Map	aimsXML.js	findXMLMode
Basic Map	aimsXML.js	hyperlinkXMLMode
Basic Map	aimsXML.js	identifyXMLMode
Basic Map	aimsXML.js	okToSend
Basic Map	aimsXML.js	pastStart
Basic Map	aimsXML.js	queryXMLMode
Basic Map	aimsXML.js	selectXMLMode
Basic Map	aimsXML.js	theImageType
Basic Map	aimsXML.js	xHalf
Basic Map	aimsXML.js	xmlEndPos
Basic Map	aimsXML.js	XMLMode
Basic Map	aimsXML.js	yHalf

## Global variables by Category (Basic Map continued)

<b>Category</b>	<b>File</b>	<b>Variable Name</b>
Basic Map	ArclMSparam.js	hostName
Basic Map	ArclMSparam.js	catURL
Basic Map	ArclMSparam.js	serverURL
Basic Map	ArclMSparam.js	imsURL
Basic Map	ArclMSparam.js	imsOVURL
Basic Map	ArclMSparam.js	imsQueryURL
Basic Map	ArclMSparam.js	imsGeocodeURL
Basic Map	ArclMSparam.js	mapBackColor
Basic Map	ArclMSparam.js	ovBoxColor
Basic Map	ArclMSparam.js	ovBoxSize
Basic Map	ArclMSparam.js	hasOVMap
Basic Map	ArclMSparam.js	hasTOC
Basic Map	ArclMSparam.js	useModeFrame
Basic Map	ArclMSparam.js	startLeft
Basic Map	ArclMSparam.js	startRight
Basic Map	ArclMSparam.js	startTop
Basic Map	ArclMSparam.js	startBottom
Basic Map	ArclMSparam.js	limitLeft
Basic Map	ArclMSparam.js	limitRight
Basic Map	ArclMSparam.js	limitTop
Basic Map	ArclMSparam.js	limitBottom
Basic Map	ArclMSparam.js	formFilePath
Basic Map	ArclMSparam.js	usePan
Basic Map	ArclMSparam.js	usePanNorth
Basic Map	ArclMSparam.js	usePanWest
Basic Map	ArclMSparam.js	usePanEast
Basic Map	ArclMSparam.js	usePanSouth
Basic Map	ArclMSparam.js	useZoomIn
Basic Map	ArclMSparam.js	useZoomOut
Basic Map	ArclMSparam.js	useFullExtent
Basic Map	ArclMSparam.js	useZoomActive
Basic Map	ArclMSparam.js	useZoomLast
Basic Map	ArclMSparam.js	useIDIdentify
Basic Map	ArclMSparam.js	useMeasure
Basic Map	ArclMSparam.js	useSetUnits

## Global variables by Category (Basic Map continued)

<b>Category</b>	<b>File</b>	<b>Variable Name</b>
Basic Map	ArcIMSpParam.js	useSelect
Basic Map	ArcIMSpParam.js	useQuery
Basic Map	ArcIMSpParam.js	useFind
Basic Map	ArcIMSpParam.js	useGeocode
Basic Map	ArcIMSpParam.js	useStoredQuery
Basic Map	ArcIMSpParam.js	useClearSelect
Basic Map	ArcIMSpParam.js	usePrint
Basic Map	ArcIMSpParam.js	useBuffer
Basic Map	ArcIMSpParam.js	useExtract
Basic Map	ArcIMSpParam.js	useHyperLink
Basic Map	ArcIMSpParam.js	useBufferShape
Basic Map	ArcIMSpParam.js	hasToolBarOnLayer
Basic Map	ArcIMSpParam.js	headerFilePath
Basic Map	ArcIMSpParam.js	footerFilePath
Basic Map	ArcIMSpParam.js	hspc
Basic Map	ArcIMSpParam.js	vspc
Basic Map	ArcIMSpParam.js	panFactor
Basic Map	ArcIMSpParam.js	zoomFactor
Basic Map	ArcIMSpParam.js	selectMargin
Basic Map	ArcIMSpParam.js	selectPointMargin
Basic Map	ArcIMSpParam.js	showScalePercent
Basic Map	ArcIMSpParam.js	showXYS
Basic Map	ArcIMSpParam.js	drawNorthArrow
Basic Map	ArcIMSpParam.js	NorthArrowType
Basic Map	ArcIMSpParam.js	NorthArrowSize
Basic Map	ArcIMSpParam.js	NorthArrowCoords
Basic Map	ArcIMSpParam.js	NorthArrowAngle
BasicMap	ArcIMSpParam.js	NorthArrowAngle
Basic Map	ArcIMSpParam.js	drawScaleBar
Basic Map	ArcIMSpParam.js	MapUnits
Basic Map	ArcIMSpParam.js	setMapUnits
Basic Map	ArcIMSpParam.js	ScaleBarUnits
Basic Map	ArcIMSpParam.js	ScaleBarBackground
Basic Map	ArcIMSpParam.js	ScaleBarBackColor
Basic Map	ArcIMSpParam.js	ScaleBarFontColor

## Global variables by Category (Basic Map continued)

<b>Category</b>	<b>File</b>	<b>Variable Name</b>
Basic Map	ArclMSparam.js	ScaleBarColor
Basic Map	ArclMSparam.js	ScaleBarFont
Basic Map	ArclMSparam.js	ScaleBarStyle
Basic Map	ArclMSparam.js	ScaleBarRound
Basic Map	ArclMSparam.js	ScaleBarSize
Basic Map	ArclMSparam.js	ScaleBarWidth
Basic Map	ArclMSparam.js	ScaleBarPrecision
Basic Map	ArclMSparam.js	numDecimals
Basic Map	ArclMSparam.js	drawCopyright
Basic Map	ArclMSparam.js	CopyrightFont
Basic Map	ArclMSparam.js	CopyrightStyle
Basic Map	ArclMSparam.js	CopyrightSize
Basic Map	ArclMSparam.js	CopyrightCoords
Basic Map	ArclMSparam.js	CopyrightColor
Basic Map	ArclMSparam.js	CopyrightBackground
Basic Map	ArclMSparam.js	CopyrightBGCOLOR
Basic Map	ArclMSparam.js	CopyrightGlow
Basic Map	ArclMSparam.js	CopyrightGlowColor
Basic Map	ArclMSparam.js	CopyrightText
Basic Map	ArclMSparam.js	drawModeOnMap
Basic Map	ArclMSparam.js	modeRefreshMap
Basic Map	ArclMSparam.js	modeMapColor
Basic Map	ArclMSparam.js	modeMapGlow
Basic Map	ArclMSparam.js	ovlImageVar
Basic Map	ArclMSparam.js	ovBorderWidth
Basic Map	ArclMSparam.js	ovExtentBoxSize
Basic Map	ArclMSparam.js	isNav4
Basic Map	ArclMSparam.js	isNav5up
Basic Map	ArclMSparam.js	isNav
Basic Map	ArclMSparam.js	isIE
Basic Map	ArclMSparam.js	ovHspc
Basic Map	ArclMSparam.js	zoomBoxColor
Basic Map	ArclMSparam.js	ActiveLayerIndex
Basic Map	ArclMSparam.js	useTextFrame
Basic Map	ArclMSparam.js	useExternalWindow

## Global variables by Category (Basic Map continued)

<b>Category</b>	<b>File</b>	<b>Variable Name</b>
Basic Map	ArcIMSpParam.js	textFrameBackColor
Basic Map	ArcIMSpParam.js	textBackColor
Basic Map	ArcIMSpParam.js	textFrameTextColor
Basic Map	ArcIMSpParam.js	textFrameLinkColor
Basic Map	ArcIMSpParam.js	textFrameFormColor
Basic Map	ArcIMSpParam.js	showTOC
Basic Map	ArcIMSpParam.js	toggleVisible
Basic Map	ArcIMSpParam.js	toggleOVVisible
Basic Map	ArcIMSpParam.js	listAllLayers
Basic Map	ArcIMSpParam.js	drawFloatingMode
Basic Map	ArcIMSpParam.js	modeLayerOn
Basic Map	ArcIMSpParam.js	modeLayerColor
Basic Map	ArcIMSpParam.js	modeLayerShadowColor
Basic Map	ArcIMSpParam.js	modeLayerFont
Basic Map	ArcIMSpParam.js	modeLayerSize
Basic Map	ArcIMSpParam.js	ovMapIsLayer
Basic Map	ArcIMSpParam.js	webParams
Basic Map	ArcIMSpParam.js	clickMarkerColor
Basic Map	ArcIMSpParam.js	clickMarkerType
Basic Map	ArcIMSpParam.js	clickMarkerSize
Basic Map	ArcIMSpParam.js	pixelTolerance
Basic Map	ArcIMSpParam.js	selectColor
Basic Map	ArcIMSpParam.js	highlightColor
Basic Map	ArcIMSpParam.js	transparentLevel
Basic Map	ArcIMSpParam.js	selectFields
Basic Map	ArcIMSpParam.js	swapSelectFields
Basic Map	ArcIMSpParam.js	selFieldList
Basic Map	ArcIMSpParam.js	useFieldAlias
Basic Map	ArcIMSpParam.js	fieldAliasList
Basic Map	ArcIMSpParam.js	hyperLinkLayers
Basic Map	ArcIMSpParam.js	hyperLinkFields
Basic Map	ArcIMSpParam.js	showSelectedData
Basic Map	ArcIMSpParam.js	showSelectedFeatures
Basic Map	ArcIMSpParam.js	maxFeaturesReturned
Basic Map	ArcIMSpParam.js	numberDataSamples

## Global variables by Category (Basic Map continued)

<b>Category</b>	<b>File</b>	<b>Variable Name</b>
Basic Map	ArclMSparam.js	numStatDecimals
Basic Map	ArclMSparam.js	legWidth
Basic Map	ArclMSparam.js	legHeight
Basic Map	ArclMSparam.js	legFont
Basic Map	ArclMSparam.js	legTitle
Basic Map	ArclMSparam.js	allowOptions
Basic Map	ArclMSparam.js	ClassRenderLayer
Basic Map	ArclMSparam.js	ClassRenderString
Basic Map	ArclMSparam.js	maxGeocodeCandidates
Basic Map	ArclMSparam.js	minGeocodeScore
Basic Map	ArclMSparam.js	geocodePointColor
Basic Map	ArclMSparam.js	geocodePointSize
Basic Map	ArclMSparam.js	geocodeLabelSize
Basic Map	ArclMSparam.js	useReverseGeocode
Basic Map	ArclMSparam.js	zoomToSingleSelect
Basic Map	ArclMSparam.js	ovVspc
Basic Map	MapFrame.htm	aimsBufferPresent
Basic Map	MapFrame.htm	aimsClassRenderPresent
Basic Map	MapFrame.htm	aimsClickPresent
Basic Map	MapFrame.htm	aimsCommonPresent
Basic Map	MapFrame.htm	aimsCustomPresent
Basic Map	MapFrame.htm	aimsDHTMLPresent
Basic Map	MapFrame.htm	aimsGenericPresent
Basic Map	MapFrame.htm	aimsGeocodePresent
Basic Map	MapFrame.htm	aimsIDIdentifyPresent
Basic Map	MapFrame.htm	aimsLayersPresent
Basic Map	MapFrame.htm	aimsLegendPresent
Basic Map	MapFrame.htm	aimsMapPresent
Basic Map	MapFrame.htm	aimsNavigationPresent
Basic Map	MapFrame.htm	aimsOptionsPresent
Basic Map	MapFrame.htm	aimsPrintPresent
Basic Map	MapFrame.htm	aimsQueryPresent
Basic Map	MapFrame.htm	aimsSelectPresent
Basic Map	Map Frame.htm	aimsXMLPresent

## Global variables by Category (Buffer-Extended Map)

<b>Category</b>	<b>File</b>	<b>Variable Name</b>
Buffer	aimsBuffer.js	bufferDistance
Buffer	aimsBuffer.js	bufferSmoothEdges
Buffer	aimsBuffer.js	bufferTargetLayer
Buffer	aimsBuffer.js	bufferTargetLayerIndex
Buffer	aimsBuffer.js	getBufferData
Buffer	aimsBuffer.js	drawTargetLayer
Extended Map	aimsLayers.js	ActiveLayer
Extended Map	aimsLayers.js	ActiveLayerType
Extended Map	aimsLayers.js	AliasFieldAliases
Extended Map	aimsLayers.js	AliasFieldNames
Extended Map	aimsLayers.js	displayLayerInfoButton
Extended Map	aimsLayers.js	FeatureLayerCount
Extended Map	aimsLayers.js	fieldIndex
Extended Map	aimsLayers.js	layersBottom
Extended Map	aimsLayers.js	layerCount
Extended Map	aimsLayers.js	LayerExtent
Extended Map	aimsLayers.js	LayerFieldCount
Extended Map	aimsLayers.js	LayerFieldList
Extended Map	aimsLayers.js	LayerFieldPrecisionList
Extended Map	aimsLayers.js	LayerFields
Extended Map	aimsLayers.js	LayerFieldSizeList
Extended Map	aimsLayers.js	LayerFieldType
Extended Map	aimsLayers.js	LayerFieldTypeList
Extended Map	aimsLayers.js	LayerID
Extended Map	aimsLayers.js	LayerIDField
Extended Map	aimsLayers.js	LayerIsFeature
Extended Map	aimsLayers.js	layersLeft
Extended Map	aimsLayers.js	LayerListOpen
Extended Map	aimsLayers.js	LayerMaxScale
Extended Map	aimsLayers.js	LayerMinScale
Extended Map	aimsLayers.js	LayerName
Extended Map	aimsLayers.js	LayerRenderString
Extended Map	aimsLayers.js	layersRight
Extended Map	aimsLayers.js	LayerShapeField
Extended Map	aimsLayers.js	layersTop

## Global variables by Category (Extended Map-Geocode)

Category	File	Variable Name
Extended Map	aimsLayers.js	queryOpen
Extended Map	aimsLayers.js	LayerType
Extended Map	aimsLayers.js	LayerVisible
Geocode	aimsGeocode.js	GCActiveLayer
Geocode	aimsGeocode.js	GAddress
Geocode	aimsGeocode.js	GCdescGeocode
Geocode	aimsGeocode.js	GCid
Geocode	aimsGeocode.js	GCidCount
Geocode	aimsGeocode.js	GClabel
Geocode	aimsGeocode.js	GCLayerCount
Geocode	aimsGeocode.js	GCLayerID
Geocode	aimsGeocode.js	GCLayers
Geocode	aimsGeocode.js	GCLayerStyle
Geocode	aimsGeocode.js	GCpointCount
Geocode	aimsGeocode.js	GCpointX
Geocode	aimsGeocode.js	GCpointY
Geocode	aimsGeocode.js	GCscore
Geocode	aimsGeocode.js	GCvalue

## Global variables by Category (IDentify/Hyperlink-Interactive)

<b>Category</b>	<b>File</b>	<b>Variable Name</b>
IDentify/Hyperlink	aimsIDentify.js	currentHyperLinkLayer
IDentify/Hyperlink	aimsIDentify.js	currentHyperLinkField
IDentify/Hyperlink	aimsIDentify.js	hyperlinkWindowWidth
IDentify/Hyperlink	aimsIDentify.js	hyperlinkWindowHeight
IDentify/Hyperlink	aimsIDentify.js	searchTolerance
IDentify/Hyperlink	aimsIDentify.js	queryStartRecord
IDentify/Hyperlink	aimsIDentify.js	canQuery
IDentify/Hyperlink	aimsIDentify.js	newSelectCount
IDentify/Hyperlink	aimsIDentify.js	selectPoints
Interactive	aimsClick.js	blankImage
Interactive	aimsClick.js	clickCount
Interactive	aimsClick.js	clickMeasure
Interactive	aimsClick.js	clickPointX
Interactive	aimsClick.js	clickPointY
Interactive	aimsClick.js	clickType
Interactive	aimsClick.js	currentMeasure
Interactive	aimsClick.js	lastToMeasure
Interactive	aimsClick.js	leftButton
Interactive	aimsClick.js	mouseX
Interactive	aimsClick.js	mouseY
Interactive	aimsClick.js	panning
Interactive	aimsClick.js	rightButton
Interactive	aimsClick.js	selectBox
Interactive	aimsClick.js	totalMeasure
Interactive	aimsClick.js	x1
Interactive	aimsClick.js	x2
Interactive	aimsClick.js	y1
Interactive	aimsClick.js	y2
Interactive	aimsClick.js	zbottom
Interactive	aimsClick.js	zleft
Interactive	aimsClick.js	zooming
Interactive	aimsClick.js	zright
Interactive	aimsClick.js	ztop

## Global variables by Category (Legend-Select)

<b>Category</b>	<b>File</b>	<b>Variable Name</b>
Legend	aimsLegend.js	defaultLegTitle
Legend	aimsLegend.js	drawLegendOnly
Print	aimsPrint.js	legVis2
Print	aimsPrint.js	printLegURL
Print	aimsPrint.js	printMapURL
Print	aimsPrint.js	printOVURL
Print	aimsPrint.js	printTitle
Query	aimsQuery.js	storedQueryCount
Query	aimsQuery.js	storedQueryFieldList
Query	aimsQuery.js	storedQueryIndex
Query	aimsQuery.js	storedQueryName
Query	aimsQuery.js	storedQueryString
Query	aimsQuery.js	storedQueryVariable
Query	aimsQuery.js	showSampleValues
Select	aimsSelect.js	highlightedOne
Select	aimsSelect.js	queryMode
Select	aimsSelect.js	selectBottom
Select	aimsSelect.js	selectBlurb
Select	aimsSelect.js	selectCount
Select	aimsSelect.js	selectData
Select	aimsSelect.js	selectEnvelope
Select	aimsSelect.js	selectionMode
Select	aimsSelect.js	selectLayer
Select	aimsSelect.js	selectLeft
Select	aimsSelect.js	selectRight
Select	aimsSelect.js	selectTop
Select	aimsSelect.js	selectType
Select	aimsSelect.js	setQueryString

## aimsBuffer.js

bufferDistance	(Numeric/Dynamic) Buffer distance. Updated by user. BufferSmoothEdges (Numeric/Dynamic) Value sent to server in buffering request based on buffer distance. Calculated by viewer prior to sending request.
bufferTargetLayer	(String/Dynamic) ID of buffer Target Layer. Updated by user on buffer form.
bufferTargetLayerIndex	(Numeric/Dynamic) Index of buffer Target Layer. Updated by viewer upon target layer selection.
drawTargetLayer	(Boolean/Dynamic) If true, draws buffer Target Layer features. Updated by buffer form.
getBufferData	(Boolean/ Dynamic) If true, attribute data will be returned for features from buffer target layer within buffer. Updated by user on buffer form.

## aimsClick.js

blankImage	(String/Static) File path of transparent image. Used with cascading style sheets/layers.
clickCount	(Numeric/Dynamic) Number of user clicks in Select Shape or Measure modes. Updated by viewer with each click.
clickMeasure	(Array/Dynamic) Array of measurements up to each click in Measure mode. Updated by viewer with each click.
clickPointX	(Array/Dynamic) Array of x-coordinates for user clicks in Select Shape or Measure modes. Updated by viewer with each click.
clickPointY	(Array/Dynamic) Array of y-coordinates for user clicks in Select Shape or Measure modes. Updated by viewer with each click.
clickType	(Numeric/Dynamic) Current click mode: 1=Measure, 2=SelectLine, 3=SelectPolygon. Updated by viewer with change of Select Shape mode or start of Measure mode.
currentMeasure	(Numeric/Dynamic) Measurement from last click to current click. Updated by viewer with each click in Measure mode.
lastTotMeasure	(Numeric/Dynamic) Last total measurement from user clicks. Updated by viewer with each click in Measure mode.
leftButton	(Numeric/Static) Left mouse button representation. Varies with browser. Updated on load by viewer.
mouseX	(Numeric/Dynamic) Map image pixel horizontal coordinate of mouse cursor. Updated by viewer on mouse movement.
mouseY	(Numeric/Dynamic) Map image pixel vertical coordinate of mouse cursor. Updated by viewer on mouse movement.
panning	(Boolean/Dynamic) If true, Pan is current mode. Checked on mouse movement. Updated by viewer on change of viewer Mode.
rightButton	(Numeric/Static) Right Mouse Button representation. Extended mapvaries with browser. Updated on load by viewer.
selectBox	(Boolean/Dynamic) If true, Select Rectangle is current mode. Checked on mouse movement. Updated by viewer on change of viewer mode.

## aimsClick.js (continued)

TotalMeasure	(Numeric/Dynamic) Total measurement from user.
x1	(Numeric/Dynamic) Map image pixel horizontal coordinate of first corner of box used in Zoom and Select Rectangle modes or starting point of Pan mode. Updated by viewer on mouse movement.
x2	(Numeric/Dynamic) Map image pixel horizontal coordinate of second corner of box used in Zoom and Select Rectangle modes or starting point of Pan mode. Updated by viewer on mouse movement.
y1	(Numeric/Dynamic) Map image pixel vertical coordinate of first corner of box used in Zoom and Select Rectangle modes or offset point of Pan mode. Updated by viewer on mouse movement.
y2	(Numeric/Dynamic) Map image pixel vertical coordinate of second corner of box used in Zoom and Select Rectangle modes or offset point of Pan mode. Updated by viewer on mouse movement.
zbottom	(Numeric/Dynamic) Map image pixel lower horizontal coordinate of box used in Zoom and Select Rectangle modes. Updated by viewer on mouse movement.
zleft	(Numeric/Dynamic) Map image pixel left vertical coordinate of box used in Zoom and Select Rectangle modes. Updated by viewer on mouse movement.
zooming	(Boolean/Dynamic) If true, one of the Zoom modes is current mode. Checked on mouse movement. Updated by viewer on change of viewer Mode.
zright	(Numeric/Dynamic) Map image pixel right vertical coordinate of box used in Zoom and Select Rectangle modes. Updated by viewer on mouse movement.
ztop	(Numeric/Dynamic) Number representing map image pixel upper horizontal coordinate of box used in Zoom and Select Rectangle modes. Updated by viewer on mouse movement.
GCActiveLayer	(String/Dynamic) ID value of current active geocode layer. Updated by viewer with each geocode request

## aimsGeocode.js

GCaddress	(Array/Dynamic) Array of strings containing the address values returned in geocode response. Updated by viewer with each geocode request.
GCdesc	(Array/Dynamic) Array of strings containing the descriptions for the geocoding input parameters to be used in geocode request. The elements in this array have corresponding elements in the GCid, GClabel, and GCvalue arrays. Updated by viewer with each geocode request.
GCid	(Array/Dynamic) Array of strings containing the id values for the geocoding input parameters to be used in geocode request. The elements in this array have corresponding elements in the GCdesc and GCvalue arrays. Updated by viewer with each geocode request.
GCidCount	(Numeric/Dynamic) Number of geocoding input parameters to be used in geocode request. Updated by viewer with start of Locate Address mode.
GClabel	(Array/Dynamic) Array of strings containing the labels for the geocoding input parameters to be used in geocode request. The elements in this array have corresponding elements in the GCid, GCdesc, and GCvalue arrays. Updated by viewer with each geocode request.
GCLayerCount	(Numeric/Dynamic) Number of geocode layers. Set by viewer on load of MapService.
GCLayerID	(Array/Dynamic) Array of strings containing the id values of layers that can be used for address matching. The elements in this array have corresponding elements in the GCLayers and GCLayerStyle arrays. Updated by viewer on load of MapService.
GCLayers	(Array/Dynamic) Array of strings containing the names of layers that can be used for address matching. The elements in this array have corresponding elements in the GCLayerID and GCLayerStyle arrays. Updated by viewer on load of MapService.
GCLayerStyle	(Array/Dynamic) Array of strings containing the geocoding style of the geocode layers. The elements in this array have corresponding elements in the GCLayers and GCLayerID arrays. Updated by viewer on load of MapService.
GCpointCount	(Numeric/Dynamic) Number of matching locations returned in geocode response. Updated by viewer with each geocode response.

## aimsGeocode.js (continued)

GCpointX	(Array/Dynamic) Array of numbers containing the x-coordinates of the matching locations. The elements in this array have corresponding elements in the GCpointY and GCscore arrays. Updated by viewer with each geocode response.
GCpointY	(Array/Dynamic) Array of numbers containing the y-coordinates of the matching locations. The elements in this array have corresponding elements in the GCpointX and GCscore arrays. Updated by viewer with each geocode response.
GCscore	(Array/Dynamic) Array of numbers containing the scores of the matching locations. The elements in this array have corresponding elements in the GCpointX and GCpointY arrays. Updated by viewer with each geocode response.
GCvalue	(Array/Dynamic) Array of strings containing the user-entered values for the geocoding input parameters to be used in geocode request. The elements in this array have corresponding elements in the GCid, GCdesc, and GClabel arrays. Updated by user on Locate Address form.

## aimsIdentify.js

currentHyperLinkLayer	(String/Dynamic) Name of current HyperLinkLayer. Updated by viewer on start of HyperLink mode.
currentHyperLinkField	(String/Dynamic) Field name of current HyperLinkLayer field. Updated by viewer on start of Hyperlink mode.
hyperLinkWindowWidth	(Numeric/Static) Width of window used to display hyperlinked URL.
hyperLinkWindowHeight	(Numeric/Static) Height of window used to display hyperlinked URL.
searchTolerance	(Numeric/Dynamic) Search tolerance. This value is calculated using pixelTolerance and extent-to-pixel ratio. Updated by viewer on identify and hyperlink requests.
queryStartRecord	(Numeric/Dynamic) Starting record position for returned records. Updated by viewer on each query/selection.
canQuery	(Boolean/Dynamic) If true, selection and query can proceed. Updated by viewer on startup.
newSelectCount	(Numeric/Dynamic) Number of current selected features. Updated by viewer with each query/selection.
selectPoints	(Array/Dynamic) Array of feature ids for selected features. Updated by viewer with each query/selection.
ActiveLayer	(String/Dynamic) ID of current active layer. Updated by viewer on change of active layer.
ActiveLayerType	(String/Dynamic) Shape type of current active layer. Updated by viewer on change of active layer.
AliasFieldName	(Array/Dynamic) Array of strings containing the names of the alias field for each layer in the map service. The elements in this array have corresponding elements in the Alias FieldAlias array. Updated by viewer on load MapService.
AliasFieldAliases	(Array/Dynamic) Array of strings containing the alias of the alias field for each layer in the MapService. The elements of this array have corresponding elements in the AliasFieldName array. Updated by viewer on load MapService.

## aimsLayers.js

displayLayerInfoButton	(Boolean/Static) If true, the LayerList will contain a button for each layer that, if clicked, will call showLayerInfo() to display layer information. Default is false.
FeatureLayerCount	(Numeric/Dynamic) Number of feature type layers in MapService. Updated by viewer on load of MapService.
fieldIndex	(Numeric/Dynamic) Index of current field of the current active layer. Updated by viewer on start of Query mode or by user in Query form.
layerCount	(Numeric/Dynamic) Number of layers in MapService. Updated by viewer on load MapService.
LayerExtent	(Array/Dynamic) Array of strings containing the extent for each layer in the MapService. The elements in this array have corresponding elements in the LayerFieldList, LayerFieldPrecisionList, LayerFieldSizeList, LayerFieldTypeList, LayerID, LayerIDField, LayerIsFeature, LayerMaxScale, LayerMinScale, LayerName, LayerRenderString, LayerShapeField, LayerType, and LayerVisible arrays. Updated by viewer on load MapService.
LayerFieldCount	(Numeric/Dynamic) Number of fields in the current active layer. Updated by viewer on change of active layer.
LayerFieldList	(Array/Dynamic) Array of strings containing lists of field names for each layer in the MapService. The elements in this array have corresponding elements in the LayerExtent, LayerFieldPrecisionList, LayerFieldSizeList, LayerFieldTypeList, LayerID, LayerIDField, LayerIsFeature, LayerMaxScale, LayerMinScale, LayerName, LayerRenderString, LayerShapeField, Layer Type, and LayerVisible arrays. Updated by viewer on load MapService.
LayerFieldPrecisionList	(Array/Dynamic) Array of strings containing lists of precisions for the fields of each layer in the MapService. The elements in this array have corresponding elements in the LayerExtent, LayerFieldList, LayerFieldSizeList, LayerFieldTypeList, LayerID, LayerIDField, LayerIsFeature, LayerMaxScale, LayerMinScale, LayerName, LayerRenderString, LayerShapeField, LayerType, and LayerVisible arrays. Updated by viewer on load MapService.
LayerFields	(Array/Dynamic) Array of strings containing the names of the fields in the current active layer. The elements in this array have corresponding elements in the LayerFieldType array. Updated by viewer on change of active layer.

## aimsLayers.js (continued)

LayerFieldSizeList	(Array/Dynamic) Array of strings containing lists of sizes for the fields of each layer in the MapService. The elements in this array have corresponding elements in the LayerExtent, LayerFieldList, LayerFieldPrecisionList, LayerFieldTypeList, LayerID, LayerIDField, LayerIsFeature, LayerMaxScale, LayerMinScale, LayerName, LayerRenderString, LayerShapeField, Layer Type, and LayerVisible arrays. Updated by viewer on load MapService.
LayerFieldType	(Array/Dynamic) Array of strings containing the types of the field in the current active layer. The elements in this array have corresponding elements in the LayerFields array. Updated by viewer on change of active layer.
LayerFieldTypeList	(Array/Dynamic) Array of strings containing lists of types for the fields of each layer in the MapService. The elements in this array have corresponding elements in the LayerExtent, LayerFieldList, LayerFieldPrecisionList, LayerFieldTypeList, LayerID, LayerIDField, LayerIsFeature, LayerMaxScale, LayerMinScale, LayerName, LayerRenderString, LayerShapeField, Layer Type, and LayerVisible arrays. Updated by viewer on load MapService.
LayerID	(Array/Dynamic) Array of strings containing the id value for each layer in the MapService. The elements in this array have corresponding elements in the LayerExtent, LayerFieldList, LayerFieldPrecisionList, LayerFieldSizeList, LayerFieldTypeList, LayerIDField, LayerIsFeature, LayerMaxScale, LayerMinScale, LayerName, LayerRenderString, LayerShapeField, Layer Type, and LayerVisible arrays. Updated by viewer on load MapService.
LayerIDField	(Array/Dynamic) Array of strings containing the names of ID fields for each layer in the MapService. The elements in this array have corresponding elements in the LayerExtent, LayerFieldList, LayerFieldPrecisionList, LayerFieldSizeList, LayerFieldTypeList, LayerID, LayerIsFeature, LayerMaxScale, LayerMinScale, LayerName, LayerRenderString, LayerShapeField, Layer Type, and LayerVisible arrays. Updated by viewer on load MapService.
LayerIsFeature	(Array/Dynamic) Array of Boolean values for each layer in the MapService indicating if it is a feature type or not. The elements in this array have corresponding elements in the LayerExtent, LayerFieldList, LayerFieldPrecisionList, LayerFieldSizeList, LayerFieldTypeList, LayerID, LayerIDField, LayerMaxScale, LayerMinScale, LayerName, LayerRenderString,

## aimsLayers.js (continued)

LayerShapeField, LayerType, and LayerVisible arrays.  
Updated by viewer on load MapService.

LayerListOpen	(Boolean/Dynamic) If true, LayerList is currently displayed in a separate window and should be updated with each map request. Updated by user by click on “Hide LayerList” or Display LayerList buttons.
LayerMaxScale	(Array/Dynamic) Array of strings containing the maximum scale threshold for each layer in the MapService. The elements in this array have corresponding elements in the LayerExtent, LayerFieldList, LayerFieldPrecisionList, LayerFieldSizeList, LayerFieldTypeList, LayerID, LayerIDField, LayerIsFeature, LayerMinScale, LayerName, LayerRenderString, LayerShapeField, LayerType, and LayerVisible arrays. Updated by viewer on load MapService.
LayerMinScale	(Array/Dynamic) Array of strings containing the minimum scale threshold for each layer in the MapService.
LayerName	(Array/Dynamic) Array of strings containing the names of the layers in the MapService.
LayerVisible	(Array/Dynamic) Array of Booleans indicating whether each of the layers is turned on or not.
LayerType	(Array/Dynamic) Array of strings containing the feature type of each layer in the MapService.
LayerRenderString	(Array/Dynamic) Array of strings containing the render information for each layer in the MapService.
LayerShapeField	(Array/Dynamic) Array of strings containing the shape field for each layer in the MapService.
layerLeft	(Numeric/Dynamic) Array of numbers representing left coordinate of the extent of each layer.
layerRight	(Numeric/Dynamic) Array of numbers representing right coordinate of the extent of each layer.
layerTop	(Numeric/Dynamic) Array of numbers representing top coordinate of the extent of each layer.

## aimsLayers.js (continued)

layerBottom	(Numeric/Dynamic) Array of numbers representing bottom coordinate of the extent of each layer.
QueryOpen	(Boolean/Dynamic) True indicates that a query is in the process of being constructed.

## aimsMap.js

appDir	(String/Static) String containing URL path of viewer files. Set by viewer on startup.
bottom	(Numeric/Dynamic) Current extent bottom (minimum y-coordinate). Updated by viewer on change of extent.
canLoad	(Boolean/Dynamic) If true, viewer can load MapServices interactively. Updated by viewer on startup. Available for custom implementation.
debugOn	(Numeric/Dynamic) Current debug mode. Available modes are: 0=0ff, 1>Show ArcXML requests sent through sendMapXML(); 2>Show ArcXML query responses; 3>Show all ArcXML requests and responses. Default is 0. Updated by custom implementation.
dQuote	(String/Static) Double-quote character.
enforceFullExtent	(Boolean/Static) If true, map extent sent in map request will always be within full extent. Set by viewer on startup.
fullBottom	(Numeric/Dynamic) Full extent bottom (minimum y-coordinate). Updated by viewer on load of MapService.
fullHeight	(Numeric/Dynamic) Full extent height. Updated by viewer on load of MapService.
fullLeft	(Numeric/Dynamic) Full extent left (minimum x- coordinate). Updated by viewer on load of MapService.
fullOVBottom	(Numeric/Dynamic) Overview map full extent bottom (minimum y-coordinate). Updated by viewer on load of MapService.
fullOVHeight	(Numeric/Dynamic) Overview map full extent height. Updated by viewer on load of MapService.
fullOVLeft	(Numeric/Dynamic) Overview map full extent left (minimum x- coordinate). Updated by viewer on load of MapService.
fullOVRRight	(Numeric/Dynamic) Overview map full extent right (maximum x- coordinate). Updated by viewer on load of MapService.
fullOVTop	(Numeric/Dynamic) Overview map full extent top (maximum y- coordinate). Updated by viewer on load of MapService.

## aimsMap.js (continued)

fullOVWidth	(Numeric/Dynamic) Overview map full extent width. Updated by viewer on load of MapService.
fullRight	(Numeric/Dynamic) Full extent right (maximum x- coordinate). Updated by viewer on load of MapService.
fullTop	(Numeric/Dynamic) Full extent top (maximum y-coordinate). Updated by viewer on load of MapService.
fullWidth	(Numeric/Dynamic) Full extent width. Updated by viewer on load of MapService.
geocodeLabel	(String/Dynamic) Label to be displayed with geocode point defined by geocodeX and geocodeY. Updated by viewer on Address match response or by zoomToPoint().
geocodeX	(Numeric/Dynamic) Geocode point x-coordinate. Updated by viewer on address match response or by zoomToPoint().
geocodeY	(Numeric/Dynamic) Geocode point y-coordinate. Updated by viewer on address match response or by zoomToPoint().
getLimitExtent	(Boolean/Dynamic) If true, viewer will request extent from server. Full and limit coordinates are set to the returned coordinates. Default value is true. If limitLeft and limitRight are both set to nonzero values, getLimitExtent will be updated to false. Updated by viewer as needed.
getStartingExtent	(Boolean/Dynamic) If true, viewer will set start coordinates using extent returned from initial GetServiceInfo requests. Default value is true. If startLeft and startRight are both set to nonzero values, getStartingExtent will be updated to false. Updated by viewer as needed.
i2Height	(Numeric/Static) Overview map image height in pixels. Set by viewer on startup.
i2Width	(Numeric/Static) Overview map image width in pixels. Set by viewer on startup.
iHeight	(Numeric/Static) Map image height in pixels. Set by viewer on startup.
imageLimitBottom	(Numeric/Dynamic) Limit extent bottom (minimum y-coordinate) of defined image size. Updated by viewer on load of MapService.

## aimsMap.js (continued)

imageLimitLeft	(Numeric/Dynamic) Limit extent left (minimum x-coordinate) of defined image size. Updated by viewer on load of MapService.
imageLimitRight	(Numeric/Dynamic) Limit extent right (maximum x-coordinate) of defined image size. Updated by viewer on load of MapService.
imageLimitTop	(Numeric/Dynamic) Limit extent top (maximum x-coordinate) of defined image size. Updated by viewer on load of MapService.
iWidth	(Numeric/Static) Map image width in pixels. Set by viewer on startup.
lastBottom	(Numeric/Dynamic) Previous extent bottom (minimum y-coordinate). Updated by viewer on change of extent.
lastLeft	(Numeric/Dynamic) Previous extent left (minimum x-coordinate). Updated by viewer on change of extent.
lastRight	(Numeric/Dynamic) Previous extent right (maximum x-coordinate). Updated by viewer on change of extent.
lastTop	(Numeric/Dynamic) Previous extent top (maximum x-coordinate). Updated by viewer on change of extent.
left	(Numeric/Dynamic) Current extent left (minimum x- coordinate). Updated by viewer on change of extent.
legendVisible	(Boolean/Dynamic) If true, map request will include request for legend image. Updated by user by clicking “Legend/LayerList” button, which calls clickFunction(“legend”).
mapScaleFactor	(Numeric/Dynamic) Ratio of extent width to image width (map units per pixel). Updated by viewer on each map request/response.
mapX	(Numeric/Dynamic) Current x-coordinate in map units. Updated by viewer on movement of cursor.
mapY	(Numeric/Dynamic) Current y-coordinate in map units. Updated by viewer on movement of cursor.
panX	(Numeric/Dynamic) Current distance in horizontal direction map will pan using arrow buttons (pan() function). Updated by viewer on change of extent using panFactor.

## aimsMap.js (continued)

panY	(Numeric/Dynamic) Current distance in vertical direction map will pan using arrow buttons (pan() function). Updated by viewer on change of extent using panFactor.
pixelX	(Numeric/Dynamic) Ratio of extent width to image width (map units per pixel). Used by viewer to convert image pixel x- coordinate to map x- coordinate. Updated by viewer on each map request/response.
pixelY	(Numeric/Dynamic) Ratio of extent height to image height (map units per pixel). Used by viewer to convert image pixel y- coordinate to map y- coordinate. Updated by viewer on each map request/response.
right	(Numeric/Dynamic) Current extent right (maximum x-coordinate). Updated by viewer on change of extent.
setDebug	(Boolean/Static) If true, debug setting can be changed. Used by custom implementation.
showGeocode	(Boolean/Dynamic) If true, map request will include command to draw point at geocodeX, geocodeY with an optional label. Updated by viewer on response of address match request or zoomToPoint() function.
sQuote	(String/Static) Single-quote character (apostrophe).
theCursor	(String/Static) Current cursor icon type. Used only on IE. Updated by viewer as needed.
toolMode	(Numeric/Dynamic) Cursor mode. Updated by user on click on toolbar buttons.
top	(Numeric/Dynamic) Current extent top (maximum y-coordinate). Updated by viewer on change of extent.
useLimitExtent	(Boolean/Static) If true, the limit extent will be enforced on map and query requests. Default value is false.
xDistance	(Numeric/Dynamic) Current map extent width. Updated by viewer on each map request/response.
yDistance	(Numeric/Dynamic) Current map extent height. Updated by viewer on each map request/response.

## aimsMap.js (defined values for toolMode)

toolMode	Description
1	Zoom In
2	Zoom Out
3	Pan
4	Identify
8	Query
9	Find
10	Select Rectangle
11	Select Point
12	Select Line
13	Select Polygon
15	Hyperlink
16	Select Shape
17	Buffer Shape
20	Measure
25	Buffer
51	StoredQuery

## aimsPrint.js

legVis2	(Boolean/Dynamic) Used by viewer to temporarily hold current legend visibility status during print mode sequence. Updated by viewer during print mode sequence.
printLegURL	(String/Dynamic) URL of Legend image to be used on print page. Updated by viewer during print mode sequence.
printMapURL	(String/Dynamic) URL of main map image to be used on print page. Updated by viewer during print mode sequence.
printOVURL	(String/Dynamic) URL of overview map image to be used on print page. Updated by viewer during print mode sequence.
printTitle	(String/Dynamic) Title to be displayed on print page. Updated by user on Print form. showSampleValues (Boolean/Dynamic). If true, Query form will display sample values for the current active field. Updated by viewer on start of Query mode or user in Query form.

## aimsQuery.js

showSampleValues	(Boolean/Dynamic) If true, Query form will display sample values for the current active field. Updated by viewer on start of Query mode or user in Query form.
storedQueryCount	(Numeric/Dynamic) Number of StoredQueries available for the current active layer. Updated by viewer on start of StoredQuery mode.
storedQueryFieldList	(Array/Dynamic) Array of strings containing a list of names of the fields to be returned in the query response for each StoredQuery available for the current active layer. Updated by viewer on start of StoredQuery mode.
storedQueryIndex	(Numeric/Dynamic) Index of current StoredQuery. This index is used in referencing elements in the storedQueryName, storedQueryString, storedQueryVariable, storedQueryVarCount, and storedQueryFieldList arrays. Updated by user in StoredQuery form.
storedQueryName	(Array/Dynamic) Array of strings containing the name of each StoredQuery available for the current active layer. Updated by viewer on start of StoredQuery mode.
storedQueryString	(Array/Dynamic) Array of strings containing the query string of each StoredQuery available for the current active layer. Updated by viewer on start of StoredQuery mode.
storedQueryVarCount	(Array/Dynamic) Array of numbers representing the count of variables used in each StoredQuery available for the current active layer. Updated by viewer on start of StoredQuery mode.
storedQueryVariable	(Array/Dynamic) Array of strings containing the variable(s) used for user-input value(s) in each StoredQuery available for the current active layer. Updated by viewer on start of StoredQuery mode.

## aimsSelect.js

highlightedOne	(String/Dynamic) Query string (if not an empty string) to be sent in map request to highlight one feature from the currently selected group of features. The format of this string is idfield = idvalue. This query string will be issued against the current active layer. The default value is an empty string. Updated by viewer on user clicking on hyperlink in display of returned record values of selected features or through custom implementation.
queryMode	(Numeric/Dynamic) Current type of query/selection process: zero=spatial selection, 1=attribute query. Updated by viewer start of appropriate query/selection mode.
selectBottom	(Array/Dynamic) Array of numbers representing the bottom coordinate of the extent of each selected feature. Updated by viewer on query/selection response.
selectBlurb	(String/Dynamic) Custom query string to be sent (if not an empty string) with map request. Query will be applied to current active layer. Default value is an empty string. Updated by custom implementation.
selectCount	(Numeric/Dynamic) Number of currently selected features. Updated by viewer on query/selection response or change of mode.
selectData	(Array/Dynamic) Array of strings containing sample data values for the query form. Updated by viewer upon user request for sample data values.
selectEnvelope	(String/Dynamic) String representation of envelope created by Select by rectangle or identify to be sent in request to server. Updated by viewer in select by rectangle and identify modes.
selectionMode	(Numeric/Dynamic) Current type of selection input: 1=query, 2=box, point; 3=line, polygon. Updated by viewer start of appropriate query/selection mode.
selectLayer	(String/Dynamic) ID of the layer that query/selection will be issued against. Updated by viewer as needed.
selectLeft	(Array/Dynamic) Array of numbers representing the left coordinate of the extent of each selected feature. Updated by viewer on query/selection response.

## aimsSelect.js (continued)

selectRight	(Array/Dynamic) Array of numbers representing the right coordinate of the extent of each selected feature. Updated by viewer on query/selection response.
selectTop	(Array/Dynamic) Array of numbers representing the top coordinate of the extent of each selected feature. Updated by viewer on query/selection response.
selectType	(String/Dynamic) The shape type of the layer that query/selection will be issued against. Updated by viewer as needed.
setQueryString	(String / Dynamic) Query string to be sent in requests to the ImageServer and QueryServer. This query string will be issued against the current active layer. Updated by viewer on query/selection request.

## aimsXML.js

drawOVExtentBox	(Boolean/Static) If true, a box defining current extent will be drawn on overview map by server. Default is false. The default viewer uses cascading style sheets (layers in Netscape) to display extent box.
findXMLMode	(Numeric/Static) Value to be used for Find requests in XMLMode.
hyperlinkXMLMode	(Numeric/Static) Value to be used for HyperLink requests in XMLMode.
identifyXMLMode	(Numeric/Static) Value to be used for IDentify requests in XMLMode.
okToSend	(Boolean/Dynamic) If true, viewer has received response from last request. Updated by viewer on each request and response.
pastStart	(Boolean/Dynamic) If true, viewer has already sent first map request for the current MapService. Updated on first map request.
queryXMLMode	(Numeric/Static) Value to be used for Query requests in XMLMode.
selectXMLMode	(Numeric/Static) Value to be used for Select requests in XMLMode.
theImageType	(String/Dynamic) Image type (JPG, GIF, PNG). Updated by viewer on each map response.
xHalf	(Numeric/Dynamic) Half of the current extent width. Updated by viewer on change of extent.
xmlEndPos	(Numeric/Dynamic) The current position to start the next parse scan of XML string. Updated by viewer as needed.
XMLMode	(Numeric/Dynamic) Number representing the current ArcXML request/response mode. Updated by viewer as needed in various functions.
yHalf	(Numeric/Dynamic) Half of the current extent height. Updated by viewer on change of extent.

## aimsXML.js (defined values)

xmlMode	Description
1	Map Image
2	Overview Map Image
3	ServiceInfo - Extent
4	ServiceInfo - Layer Info
5	List of MapServices
6	Select
7	Identify
8	Query
9	Geocode
11	Buffer
14	Find
15	Hyperlink
25	Get Geocode Layers
26	Get Geocode Parameters
27	Geocode Request
40	Get List of sample field values
55	Get Layer StoredQueries
70	Get Layer Field for submission to external DB
98	Map & Legend Images
99	Map image only
101	Map image for Print routine
102	Overview image for Print routine
103	Legend image for Print routine
900	Map image only
901	Map image only with clicks
902	Overview map image only
999	Blank Map image to get modified extent

## ArcIMSpParam.js

The global variables are listed here in the order they appear in the ArcIMSpParam.js file.

hostName	(String/Dynamic) Name of machine where web document originated. Set by the user on startup.
catURL	(String/Dynamic) URL to request catalog of MapServices using hostName. Set by the user on startup.
serverURL	(String/Dynamic) String containing MapService URL prefix used in loading MapService Images.

The following variables are written by ArcIMS Designer.

imsURL	(String/Dynamic) URL of Image MapService used in main map Display.
imsOVURL	(String/Dynamic) URL of Image MapService used in overview map Display. If value is "" (empty quotes), value of imsURL is used here.
imsQueryURL	(String / Dynamic) URL of MapService used for querying. If value is "" (empty quotes), value of imsURL is used here appended with "&CustomService=Query".
imsGeocodeURL	(String/Dynamic) URL of MapService used for geocoding. If value is "" (empty quotes), value of imsURL is used here appended with "&CustomService=Geocode".
mapBackColor	(String/Dynamic) Background color of the map display expressed as an R, G, B value. If value is "" (empty quotes), color defined in AXL file is used.
ovBoxColor	(String/Dynamic) Color of Extent Box in overview map expressed as an R, G, B value. A value is required.
ovBoxSize	(Numeric/Dynamic) Size of Extent Box in overview map and ZoomBox in main map Display. A value is required.
hasOVMap	(Boolean/Dynamic) If true, the viewer has an overview map.
hasTOC	(Boolean/Dynamic) If true, the viewer has a LayerList Display.
useModeFrame	(Boolean/Dynamic) If true, the viewer displays current Tool Mode in separate frame.

## ArclMSparam.js (continued)

The following set of variables beginning with “start” represent the initial map extent values. If these are set to zero, the initial extent is set to the extent saved in MapService axl file.

startLeft	(Numeric/Dynamic) Minimum X coordinate of the map extent.
startRight	(Numeric/Dynamic) Maximum X coordinate of the map extent.
startTop	(Numeric/Dynamic) Maximum Y coordinate of the map extent.
startBottom	(Numeric/Dynamic) Minimum Y coordinate of the map extent.

The following set of variables beginning with “limit” represent map extent limit values. If these are set to zero, the extent limit is set to the map extent saved in the MapService axl file.

limitLeft	(Numeric/Dynamic) Minimum X coordinate of the map extent limit.
limitRight	(Numeric/Dynamic) Maximum X coordinate of the map extent limit.
limitTop	(Numeric/Dynamic) Maximum Y coordinate of the map extent limit.
limitBottom	(Numeric/Dynamic) Minimum Y coordinate of the map extent limit.
formFilePath	(String/Static) File path to Form files needed by the servlet connector to return response within a form. This is usually the file path to the site directory.

The following set of variables beginning with “use” are all (Boolean/Dynamic). Each sets whether or not the tool is displayed on toolbar. The function checkparams( ) confirms that the required JavaScript files are loaded. If not, the value is set to false, and the tool is not displayed on the toolbar.

usePan	If true, displays pan button in the toolbar
usePanNorth	If true, displays panNorth button in the toolbar
usePanWest	If true, displays panWest button in the toolbar
usePanEast	If true, displays panEast button in the toolbar
usePanSouth	If true, displays panSouth button in the toolbar
useZoomIn	If true, displays ZoomIn button in the toolbar
useZoomOut	If true, displays ZoomOut button in the toolbar
useFullExtent	If true, displays FullExtent button in the toolbar
useZoomActive	If true, displays ZoomActive button in the toolbar
useZoomLast	If true, displays ZoomLast button in the toolbar
useIDIdentify	If true, displays IDIdentify button in the toolbar
useMeasure	If true, displays Measure button in the toolbar
useSetUnits	If true, displays SetUnits button in the toolbar
useSelect	If true, displays Select button in the toolbar
useQuery	If true, displays Query button in the toolbar
useFind	If true, displays Find button in the toolbar
useGeocode	If true, displays Geocode button in the toolbar
useStoredQuery	If true, displays Stored Query button in the toolbar
useClearSelect	If true, displays Clear Select button in the toolbar
usePrint	If true, displays Print button in the toolbar

## ArcIMSpParam.js (continued)

useBuffer	If true, displays buffer button in the toolbar
useExtract	If true, displays Extract button in the toolbar. Extract currently not implemented.

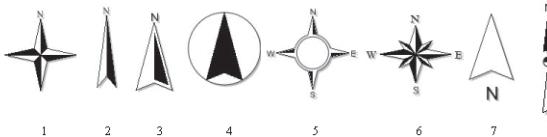
**This is the end of the variables set by ArcIMS Designer.**

useHyperLink	(Boolean/Dynamic) If true, this function is available and a button will be displayed in the default toolbar.
useBufferShape	(Boolean/Dynamic) Not implemented.
hasToolBarOnLayer	(Boolean/Static) If true, the Toolbar is on a cascading style sheet (Netscape's layer) on the Map's frame. Requires layerList.js, or appropriate function to create Toolbar.
headerFilePath	(String/Static) File path to the header HTML file used to create the submission form used by the viewer.
footerFilePath	(String/Static) File path to the footer HTML file used to create the submission form used by the viewer.

### Basic Map parameters:

hspc	(Numeric/Static) Horizontal offset (in pixels) of the Map image. This offset is from the left edge of the MapFrame page.
vspc	(Numeric/Static) Vertical offset (in pixels) of the Map image. This offset is from the top edge of the MapFrame page.
panFactor	(Numeric/Static) Pan factor used when one of the arrow pan buttons are clicked. This factor is multiplied by the current map extent and the map is panned by the resulting value.
zoomFactor	(Numeric/Static) Zoom factor used when a simple zoom is called. This factor is multiplied by the current map extent and the map is zoomed by the resulting value.
selectMargin	(Numeric/Static) Margin factor for zooming in on selected lines and polygons. Margin is this value * width and height of selected feature
selectPointMargin	(Numeric/Static) Margin factor for zooming in on selected points based on full extent. Margin is this value * full extent.

## ArcIMSpParam.js (continued)

showScalePercent	(Boolean/Dynamic) If true, display the current scale factor in the status bar when showXYS is true.
showXYS	(Boolean/Dynamic) If true, the current cursor coordinates are displayed in the browser.
drawNorthArrow	(Boolean/Static) If true, a North arrow is drawn on the Map.
NorthArrowType	(String/Static) Values 1-8 indicating the graphic style of the North arrow.
	
NorthArrowSize	(String/Static) Size of North arrow in pixels.
NorthArrowCoords	(String/Static) Screen coordinates in pixels with 0,0 at lower left corner.
NorthArrowAngle	(String/Static) Angle to rotate North arrow in counter-clockwise direction. A value of 0 is pointing right or "East", 90 is pointing up or "North".
drawScaleBar	(Boolean/Static) If true, a Scale Bar is drawn on the Map.
MapUnits	(String/Dynamic) Type of units of the Map. These are DEGREES, FEET, or METERS. Can be updated by user if useSetUnits and setMapUnits are true.
setMapUnits	(Boolean/Static) If true, and useSetUnits is true, MapUnits value be changed by user.
ScaleBarUnits	(String/Dynamic) Type of units for the ScaleBar, buffer, and Measure display. Possible values are FEET, METERS, MILES, or KILOMETERS. Can be updated by user if useSetUnits is true.
ScaleBarBackground	(Boolean/Static) If true, text has a background.
ScaleBarBackColor	(String/Static) Color of the text background expressed as an R,G,B value.
ScaleBarFontColor	(String/Static) Color of the text font expressed as an R,G,B value.
ScaleBarColor	(String/Static) Color of the bar expressed as an R,G,B value.
ScaleBarFont	(String/Static) The text font face. This value is case-sensitive.
ScaleBarStyle	(String/Static) The text font style.
ScaleBarRound	(String/Static) Number of digits to round.

## ArcIMSparam.js (continued)

ScaleBarSize	(String/Static) Length of bar in pixels.
ScaleBarWidth	(String/Static) Width of bar in pixels.
ScaleBarPrecision	(String/Static) Number of decimal places.
numDecimals	(Numeric/Dynamic) Number of decimals displayed in ScaleBar, Measure, and Coordinate display.
drawCopyright	(Boolean/Static) If true, a Copyright blurb is drawn on the map.
CopyrightFont	(String/Static) The text font face. This value is case-sensitive.
CopyrightStyle	(String/Static) The text font style.
CopyrightSize	(String/Static) Size of font in pixels.
CopyrightCoords	(String/Static) Coordinate location of text in pixels with 0,0 at lower left corner.
CopyrightColor	(String/Static) Color of the text expressed as an R,G,B value.
CopyrightBackground	(Boolean/Static) If true, a filled box behind the text is displayed.
CopyrightBGColor	(String/Static) Color of the background box expressed as an R,G,B value.
CopyrightGlow	(Boolean/Static) If true, text has a glow effect.
CopyrightGlowColor	(String/Static) Color of the glow expressed as an R,G,B value.
CopyrightText	(String/Static) Text string.
drawModeOnMap	(Boolean/Static) If true, current Tool Mode is drawn on the map.
modeRefreshMap	(Boolean/Dynamic) If true, and drawModeOnMap is true, the map Display image is refreshed with change of Tool Modes. If this value is false, the image is not refreshed until next request for new map image.
modeMapColor	(String/Static) Color of the mode text expressed as an R,G,B value.
modeMapGlow	(String/Static) Color of the glow expressed as an R,G,B value.
ovImageVar	(String/Dynamic) Overview map Image name. Updated by viewer on startup.
ovBorderWidth	(Numeric/Static) Width of border around overview map image.
ovExtentBoxSize	(Numeric/Static) Width of lines used to show current extent in overview map image.
isNav4	(Boolean/Dynamic) If true, Netscape Navigator version 4 is being used.
isNav5up	(Boolean/Dynamic) If true, Netscape Navigator version 5 is being used.
isNav	(Boolean/Dynamic) If true, Netscape Navigator is being used.
isIE	(Boolean/Dynamic) If true, Microsoft Internet Explorer is being used.

### Extended Map parameters:

## ArclMSparam.js (continued)

ovHspc	(Numeric/Static) Horizontal offset (in pixels) of the overview map image. This offset is from the left edge of the MapFrame page.
ovVspc	(Numeric/Static) Vertical offset (in pixels) of the overview map image. This offset is from the top edge of the MapFrame page.
zoomBoxColor	(String/Dynamic) Color for main map Display ZoomBox expressed in either HTML hexadecimal RGB format (“#rrggbb”) or standard HTML color name, for example, “white”, “purple”, or “red”.
ActiveLayerIndex	(Numeric/Dynamic) Index of initial Active Layer. If this value is equal to or larger than the layer count, then the top layer (0) is used. Subsequently updated by viewer on change of Active Layer.
useTextFrame	(Boolean/Dynamic) If true, there is a frame called TextFrame for data display. Updated by viewer on startup.
useExternalWidow	(Boolean/Dynamic) If true, send all data display to another browser window. If there is no TextFrame, this is set to true by the viewer.

The following set of variables set text or table parameters for the information in the TextFrame location. Colors are expressed in either HTML hexadecimal RGB format (“#rrggbb”) or standard HTML color name, for example, “white”, “purple”, or “red”.

textFrameBackColor	(String/Static) Color of the background of the page.
tableBackColor	(String/Static) Color of the cells in the table.
textFrameTextColor	(String/Static) Color of the text in the table
textFrameLinkColor	(String/Static) Color of the links in the table
textFrameFormColor	(String/Static) Color of form background.
showTOC	(Boolean/Static) If true, the LayerList will be visible on loading.
toggleVisible	(Boolean/Static) If true, layer visibility can be set by the LayerList or by custom programming.
toggleOVVisible	(Boolean/Static) If true, layer visibility in the overview map can be set by the LayerList or by custom programming the value variable. imsURL must be the same as imsOVURL.
listAllLayers	(Boolean/Dynamic) If true, the LayerList will show all layers, not just those available at current map scale.

## ArcIMSparam.js (continued)

drawFloatingMode	(Boolean/Static) If true, current Tool Mode will be drawn on a style sheet (Netscape layer) in MapFrame frame (DHTML display).
modeLayerOn	(Boolean /Dynamic) If true, current status of displaying Tool Mode on a style sheet (Netscape layer) in MapFrame frame (DHTML display).

The following set of variables beginning with “mode” set parameters for the drawing Tool Mode. These values are used in writing DHTML display.

modeLayerColor	(String/Static) Color of the text expressed in either HTML hexadecimal RGB format (“#rrggbb”) or standard HTML color name, for example, “white”, “purple”, or “red”.
modeLayerShadowColor	(String/Static) Color of the shadows expressed in either HTML hexadecimal RGB format (“#rrggbb”) or standard HTML color name, for example, “white”, “purple”, or “red”.
modeLayerFont	(String/Static) The text font face. This value is case-sensitive.
modeLayerSize	(String/Static) Size of font in pixels.
ovMapIsLayer	(Boolean/Static) If true, the overview map is an inset layer in the upper-left corner of the main map (the default configuration). If false, it is assumed that the overview map does not overlap the main map.
webParams	(String/Static) Parameters in viewer URL.

The following set of variables beginning with “click” set parameters for the drawing drawing point markers at the click location when Measure, Shape Selection, or buffer are performed.

clickMarkerColor	(String/Static) Color of the marker expressed as an R,G,B value.
clickMarkerType	(String/Static) Shape of marker.
clickMarkerSize	(String/Static) Size of marker in pixels.

### IDeIdentify>Select/Query/Buffer parameters:

pixelTolerance	(Numeric/Dynamic) Search tolerance in pixels around click.
selectColor	(String/Dynamic) Color for Selected Features expressed as an R,G,B value.
highlightColor	(String/Dynamic) Color for Highlighted Feature expressed as an R,G,B value.
transparentLevel	(String/Dynamic) Transparency level of fill color used to display selected and highlighted features.

## ArclMSparam.js (continued)

zoomToSingleSelect	(Boolean/Dynamic) If true and only one feature is returned in a query/selection, then viewer will zoom in to area surrounding selected feature.
selectFields	(String/Dynamic) A list of fields to be returned in identify/selection/query request... #ALL#=all fields. If not #ALL#, then ID and shape fields are required in list (i.e., selectFields= "NAME #ID# #SHAPE# AREA");). If swapSelectFields is true, this is updated by viewer on change of active layer with appropriate element in selFieldList array.
swapSelectFields	(Boolean/Static) If true, viewer will swap out the list of returned fields for each layer; also, a list must be defined in selFieldList array for each layer to be used in updating selectFields.
selFieldList	(Array/Dynamic) Array of strings containing a list of fields to be Returned in query/selection response for each layer. If swapSelectFields is true, these elements will be used to update selectFields. A list must be defined for each layer. Use "#ALL# for all fields.  selFieldList[0]="#ALL#"; selFieldList[1]="NAME #ID# #SHAPE# POP"; selFieldList[2]="#ALL#"; selFieldList[3]="#ALL#"; selFieldList[4]="#ALL#";  Sample set. In this case, Layer 1 will only have the fields NAME, #ID#, #SHAPE#, and AREA returned. All other layers will have all fields returned. Note: if not #ALL#, then ID and shape fields are required. Field names are separated by spaces.
useFieldAlias	(Boolean/Static) If true, viewer will use the field alias for column heading in the display of the returned field values. A list of field names and aliases must be defined in fieldAliasList array for each layer to be used in displaying the aliases as column headers.
fieldAliasList	(Array/Dynamic) Array of strings containing a list of field names and the aliases to be used as column header in the display of field values returned in query/selection response for each layer. If useFieldAlias is true, these elements will be used as aliases for corresponding field names. A list must be defined for each layer. Use "" (empty string) if the layer will have no aliases.  fieldAliasList [0]="";

## ArcIMSpParam.js (continued)

```
fieldAliasList [1] = "NAME:City Name|POP:Population";
fieldAliasList [2] = "";
fieldAliasList [3] = "";
fieldAliasList [4] = "";
```

Sample set. In this case, Layer 1 will have the columns displaying values from the fields NAME and POP will have the aliases City Name and Population used as column headers. All other columns will use the field names. All the other layers will not use any aliases. Field name/alias pairs are separated by a bar [|]. The field name and its alias are separated by a colon (:).

hyperLinkLayers	(Array/Dynamic) Array of strings containing the names of the layers that have field values to be used as hyperlinks or “HotLinks”. Each element must have a corresponding element in the hyperLinkFields array.
hyperLinkFields	(Array/Dynamic) Array of strings containing the names of the fields that have values to be used as hyperlinks or “HotLinks”. Each element must have a corresponding element in the hyperLinkLayers array.
	hyperLinkLayers[0] = “Image”; hyperLinkFields[0] = “URL”;
	Sample set. In this case, the layer called “Image” will have a hyperlink when the value from the field “URL” is displayed. The link’s URL will be the field’s value. Match is case-sensitive.
showSelectedData	(Boolean/Static) If true, the data from selected features will be displayed.
showSelectedFeatures	(Boolean/Static) If true, the selected features will be drawn.
maxFeaturesReturned	(Numeric/Static) Maximum number of records returned from query/selection.
numberDataSamples	(Numeric/Static) Number of sample records returned to display sample values in Query form.

### Legend parameters:

legWidth (Numeric/Static) Width of the legend image in pixels.

## ArclMSparam.js (continued)

legHeight	(Numeric/Static) Height of the legend image in pixels.
legFont	(String/Static) The text font face. This value is case-sensitive.
legTitle	(String/Dynamic) Title text displayed on the legend.

### Options parameters:

(Requires custom JavaScript library file aimsOptions.js, found in HTML Viewer Samples)

allowOptions	(Boolean/Static) If true, an Options page is available. When set to false, the Options button will not be displayed and the user will not be able to set options.
--------------	---

### ClassRender parameters:

These override the layer's default renderer. A sample custom JavaScript library aimsClassRender.js, found in the HTML Viewer Samples, can be used to interactively change these values.

ClassRenderLayer	(Array/Dynamic) Array of layer names for custom class rendering. Overrides default renderer. Each element must have corresponding element in the ClassRenderString array.
------------------	---

ClassRenderString	(Array/Dynamic) Array of custom render parameters for custom class rendering. Overrides default renderer. Each element must have corresponding element in the ClassRenderLayer array.
-------------------	---

ClassRenderLayer [0]="" Cities";  
ClassRenderString [0]=""

sample custom setup. If string is not empty string, default rendering is overridden with string.

numStatDecimals	(Numeric/Dynamic) Number of decimals used in statistical calculations.
-----------------	--

### Geocode parameters:

maxGeocodeCandidates	(Numeric/Static) Number representing maximum number of candidates to be returned from address match.
----------------------	--

minGeocodeScore	(Numeric/Static) Number representing minimum acceptable geocode score for a returned candidate for address match.
-----------------	---

## ArcIMSpParam.js (continued)

**Parameters to be added to map request for drawing location point for each geocode Point.**

geocodePointColor	(String/Dynamic) Color of the geocoded point expressed as an R,G,B value.
geocodePointSize	(String/Dynamic) Size of geocoded point in pixels.
geocodeLabelSize	(String/Dynamic) Size of label in pixels.
useReverseGeocode	(Boolean/Dynamic) Boolean flag for enabling reverse geocoding when implemented. Must be false unless custom implementation has been installed.

## MapFrame.htm

aimsBufferPresent	(Boolean/Static) If true, the Buffer script has been loaded.
aimsClassRenderPresent	(Boolean/Static) If true, the ClassRender script has been loaded. Located in Samples.
aimsClickPresent	(Boolean/Static) If true, the Click script has been loaded.
aimsCommonPresent	(Boolean/Static) If true, the Common script has been loaded.
aimsCustomPresent	(Boolean/Static) If true, the Custom script has been loaded.
aimsDHTMLPresent	(Boolean/Static) If true, the DHTML script has been loaded.
aimsGenericPresent	(Boolean/Static) If true, the Generic script has been loaded. Located in Samples.
aimsGeocodePresent	(Boolean/Static) If true, the Geocode script has been loaded.
aimsIdentifyPresent	(Boolean/Static) If true, the Identify script has been loaded.
aimsLayersPresent	(Boolean/Static) If true, the Layers script has been loaded.
aimsLegendPresent	(Boolean/Static) If true, the Legend script has been loaded.
aimsMapPresent	(Boolean/Static) If true, the Map script has been loaded.
aimsNavigationPresent	(Boolean/Static) If true, the Navigation script has been loaded.
aimsOptionsPresent	(Boolean/Static) If true, the Options script has been loaded. Located in Samples.
aimsPrintPresent	(Boolean/Static) If true, the Print script has been loaded.
aimsQueryPresent	(Boolean/Static) If true, the Query script has been loaded.
aimsSelectPresent	(Boolean /Static) If true, the Select script has been loaded.
aimsXMLPresent	(Boolean/Static) If true, the XML script has been loaded.
cornerOffset	(Numeric/Static) Off set from corner of style sheet (Netscape Layer).
displayLayerInfoButton	(Boolean/Static) If true, an Info button is displayed in the LayerList for each layer. Default value is false.

## MapFrame.htm (continued)

ovIsVisible	(Boolean/Dynamic) If true, the style sheets (Netscape Layers) that contain the overview map are visible.
thePageDoc	(Object/Static) Document object. Used in determining size of frame in Internet Explorer browsers.
thePageWin	(Object/Static) Window object. Used in determining size of frame in Netscape browsers.