

ArcIMS[®] 9

Customizing the HTML Viewer



Copyright © 2004 ESRI
All Rights Reserved.
Printed in the United States of America.

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

ESRI, ArcIMS, GIS by ESRI, ArcMap, ArcPad, ArcExplorer, www.esri.com, and the ArcGIS logo are trademarks, registered trademarks, or servicemarks of ESRI in the United States, the European Community, or certain other jurisdictions.

Other companies and products mentioned herein are trademarks or registered trademarks of their respective trademark owners.

Contents

1 Introducing the HTML Viewer 1

- What's new in 9.0 2
- HTML Viewer files and how they work 3
- Multiple hosts 8

2 Customizing the HTML Viewer 11

- Working with ArcIMSParam.js 12
- Working with services and map layout 14
- Working with an acetate layer 18
- Modifying attribute data display 20
- Changing the title, logo, and colors 22
- Working with tools and the toolbar 23
- Creating hyperlinks 25
- How the viewer and server communicate 26
- JavaScript function files 32
- Using the sample HTML Viewers 33

3 HTML Viewer JavaScript Library 39

- Organization of the HTML Viewer JavaScript Library 40
- JavaScript functions by name 43
- JavaScript functions by category 49
 - addBufferToMap 55
 - addCustomToMap1, addCustomToMap2, addCustomToMap3, addCustomToMap4 56
 - addLegendToMap 57
 - addSelectToMap 58
 - afterMapRefresh 59
 - beforeMapRefresh 60
 - boxIt 61
 - bufferIt 62
 - calcDistance 63
 - calcSelectEnvelope 64
 - checkCoords 65
 - checkCurrentExtent 66
 - checkForForbiddenTags 67
 - checkFullExtent 68
 - checkHyperLinkLayer 69
 - checkIfActiveLayerAvailable 70
 - checkParams 71
 - checkSelected 72
 - checkStoredQueries 73
 - chkMouseUp 74
 - clearError 75
 - clearLeadingSpace 76
 - clearSelection 77
 - clickAddPoint 78
 - clickFunction 79
 - clipLayer 80
 - compare 81

convertDecimal 82
convertHexToDec 83
convertUnits 84
createLayer 85
customMapTool 86
deleteClick 87
displayAttributeDataforDrill 88
doIdentifyAll 89
extractIt 90
findForm 91
fixSingleQuotes 92
formatDate 93
fullExtent 94
getAllFieldValues 95
getBufferAttributeData 96
getCommandLineParams 97
getEnvelopeXYs 98
getFieldNames 99
getFieldValues 100
getFind 101
getGeocodeLayers 102
getGeocodeParams 103
getHost 104
getIDValue 105
getImageXY 106
getInsideString 107
getLayer 108
getLayerFieldNames 109
getLayerFieldPrecisions 110
getLayerFieldSizes 111
getLayerFieldTypes 112
getLayers 113
getLegend 114
getLegendURL 115
getMapHeight 116
getMapWidth 117
getMapXY 118
getMoreBufferData 119
getMoreData 120
getMouse 121
getOVImageXY 122
getOVXYs 123
getPath 124
getPrintLegend 125
getPrintMap 126
getPrintOV 127
getScaleBarDistance 128
getService 129
getStartExtent 130
getStoredQueries 131
getURL 132
getXMLErrorMessage 133

- getXYs 134
- hasLayer 135
- hideLayer 136
- hideRetrieveData 137
- hideRetrieveMap 138
- htmlSendToServer 139
- hyperLink 140
- hyperLinkAny 141
- identify 142
- identifyAll 143
- isNotSameHostInURL 144
- isVisible 145
- jspSendToServer 146
- justGetFeatureCount 147
- justGetFieldValue 148
- justGetMap 149
- justGetValue 150
- makeXMLsafe 151
- mapTool 152
- moveLayer 153
- numberorder 154
- ovMapClick 155
- ovMap2Click 156
- pan 157
- panButton 158
- panMouse 159
- parseEntity 160
- parseFieldSamples 161
- parseFieldSamplesUnique 162
- parseGeocodeLayers 163
- parseGeocodeParams 164
- parseGeocodeResults 165
- parseHyperLink 166
- parseHyperLinkAny 167
- parseRecordString 168
- parseStartQuery 169
- parseStoredQueries 170
- printIt 171
- processStartExtent 172
- processXML 173
- putExtentOnOVMap 174
- queryForm 175
- reloadApp 176
- replaceLayerContent 177
- replacePlus 178
- resetClick 179
- resetError 180
- resetPanImage 181
- saveLastExtent 182
- select 183
- sendCustomToServer 184
- sendMapXML 185
- sendQueryString 186

sendShapeSelect 187
sendStoredQuery 188
sendToServer 189
setActiveLayer 190
setClip 191
setExtent 192
setFullExtent 193
setLayerBackgroundColor 194
setLayerFields 195
setStartQuery 196
setupGeocode 197
setupLayerVisible 198
setZoomColor 199
showHighlight 200
showLayer 201
showLayerInfo 202
showLegend 203
showRetrieveData 204
showRetrieveMap 205
startExtent 206
startMap 207
startPan 208
startSelectBox 209
startUp 210
startZoomBox 211
startZoomOutBox 212
stopPan 213
stopSelectBox 214
stopZoomBox 215
stopZoomOutBox 216
storedQueryForm 217
swapQuotes 218
swapStuff 219
tempGetSamples 220
toggleOVMap 221
untag 222
updateMeasureBox 223
useCustomFunction 224
writeBlankMapXML 225
writeBufferForm 226
writeEnvelopeBufferXML 227
writeEnvelopeXML 228
writeFieldSample 229
writeFindRequest 230
writeGeocodeXML 231
writeGetBufferedData 232
writeGetBufferFeatures 233
writeGetFeatures 234
writeGetFeatures2 235
writeGetFeatures3 236
writeGetFeaturesDrill 237
writeIdentifyXML 238
writeLayerListForm 239

writeModeFrame	240
writeModeLayers	241
writeOutDataPage	242
writeOVXML	243
writePrintPage	244
writeQueryBufferXML	245
writeQueryForm	246
writeQueryXML	247
writeShapeBufferXML	248
writeShapeSelect	249
writeStartQueryXML	250
writeStoredQueryForm	251
writeXML	252
zoomBack	253
zoomButton	254
zoomin	255
zoomout	256
zoomScale	257
zoomToEnvelope	258
zoomToPoint	259
zoomToReturnedRecords	260
Global variables by name	261
Global variables by category	273
aimsBuffer.js	285
aimsClick.js	286
aimsCommon.js	288
aimsGeocode.js	289
aimsIdentify.js	291
aimsLayers.js	293
aimsLegend.js	296
aimsMap.js	297
aimsMap.js (defined values for toolMode)	300
aimsPrint.js	301
aimsQuery.js	302
aimsResource.js	303
aimsSelect.js	304
aimsXML.js	306
aimsXML.js (defined values)	308
ArcIMSparam.js	309
MapFrame.htm	320

Introducing the HTML Viewer

1

IN THIS CHAPTER

- **What's new in 9.0**
- **HTML Viewer files and how they work**
- **Multiple hosts**

The HTML Viewer, a viewer installed automatically with ESRI® ArcIMS® software, defines the graphical look and functionality of your ArcIMS Web site or sites. Using HTML, a significant amount of JavaScript™, and some Dynamic HTML (DHTML), the viewer provides a framework for the map, toolbar, legend, overview map, and other graphic elements for the site. HTML, JavaScript, and DHTML are increasingly being used by Web developers to add flexibility and interactivity to their Web pages.

The default HTML Viewer is a set of HTML pages and JavaScript files that you create in ArcIMS Designer when you design your site. Although the Designer gives you many design options, you may want more choices. By customizing the default HTML Viewer, you get many more choices. For example, you can:

- Insert your own company logo.
- Add functionality.
- Change the frame layout.
- Modify the toolbar.
- Change the graphic look.

The HTML Viewer has been carefully designed and tested so that its elements and functions display and behave appropriately in a wide range of Web browsers and versions of them. Changes to the viewer may affect this cross browser accommodation.

This guide explains the foundation for customizing the HTML Viewer and provides a complete reference to the function library available with ArcIMS. The guide is intended for users who have an understanding of what an ArcIMS viewer is, a working knowledge of HTML, and familiarity with JavaScript.

What's new in 9.0

The following functions have been added to the HTML Viewer:

- In `aimsBuffer.js`, the `getMoreBufferData(startRecord)` and `writeGetBufferFeatures()` functions
- In `aimsIdentify.js`, the `writeOutDataPage(theReply,layerIndex,docObject,docName,pageObject)` function
- In `aimsXML.js`, the `resetPanImage()` function

The following files are no longer part of the viewer:

- `displayAttributeData.htm`
- `displayAttributeDataAll.htm`
- `displayBufferAttributeData.htm`

Two viewer sample customizations that came with ArcIMS version 4.0.1 relied on a .dll file that was retired at ArcIMS 9.0. One of them, the Tracker sample viewer, has been removed completely from ArcIMS. The other, the Parcels sample viewer, is still available, but its VB portion has been removed from ArcIMS.

The following are additions to HTML Viewer documentation:

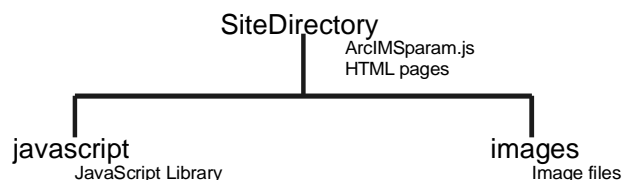
- Instructions for using the Java Connector with the HTML Viewer to prevent end users with ArcMap™ from accessing your source data. See the 'Multiple hosts' section in Chapter 1 of this guide for more information.
- Details for using the GET_SERVICE_INFO request with the HTML Viewer, for example, how to load layer parameters from file to decrease viewer startup time. See 'How the viewer and server communicate' in Chapter 2 of this guide for more information.

HTML Viewer files and how they work

ArcIMS Designer creates output files, including HTML and JavaScript files (.js), that form the foundation of the HTML Viewer. It creates them in the directory structure described in this section. HTML files are used to generate Web page content, and the JavaScript functions allow for user interaction with the map.

Directory structure

When you create a Web site using the default HTML Viewer through Designer, a hierarchy of directories and files is created under the directory you specified as your Web site directory when installing ArcIMS. 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 subdirectory to reduce the amount of storage space the HTML Viewer uses. Note that deleting this subdirectory makes the directory smaller in size but does not affect viewer initialization (it does not affect the browser’s loading of the viewer).

The ArcIMSParam.js parameters file

Located in the Web site directory is a parameters file named `ArcIMSParam.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 `ArcIMSParam.js` file.

The HTML files

There are approximately 30 HTML files that define the page content for the HTML Viewer. Some are used to initialize the viewer, some define the viewer’s frames content, and others are used to load a form and display data.

File sequence during initialization

When opening an HTML Viewer created by Designer, files are opened and accessed in a sequence. This is known as viewer initialization, startup, or preloading. The following steps describe this sequence in order. You may be able to decrease viewer startup time by specifying that you want to load the layer parameters from file rather than by parsing them from the `GET_SERVICE_INFO` response. For more information, see the ArcXML request/response cycle section in Chapter 2, ‘Customizing the HTML Viewer’.

1. Default.htm is the entry point into your Web site. This file checks if the client's browser is either Microsoft® Internet Explorer or Netscape®, version 5 or newer (5.1 or later on Macintosh IE), or another 5.x compatible browser such as Mozilla or Opera. If so, Run.htm is loaded, which defines two frames for Authorize.htm and jsForm.htm. If it is not a supported browser, CannotRun.htm is loaded and informs the user.
2. Authorize.htm checks if security measures for the services have been set up. If so, the login dialog is opened, and the user is required to log in. Authorize.htm also calls two JavaScript files: ArcIMSParam.js and aimsResource.js. A GET_SERVICE_INFO request is sent for the main map. If the overview map does not use the same service as the main map, a separate GET_SERVICE_INFO request is sent to check for the existence and verification of any user authentication requirements for that service. ArcIMSParam.js contains various parameters for the viewer including the URLs of the services used for the main and overview map. AimsResource.js contains the various text displays used by messages, buttons, and titles.
3. The viewer page (viewer.htm) is loaded.

Viewer.htm defines the frame layout of the viewer. The frames provide structure to the layout by dividing up the Web page into multiple “sections”. Each frame is filled with an HTML file. There are frames for the main map display, the overview map, the layer list, and toolbar, along with a few others. While some of the frames can be removed, the viewer requires MapFrame and PostFrame be present.
4. MapFrame.htm and PostFrame.htm are then loaded into their appropriate frames. Other HTML files that provide frame content are listed later in this chapter under ‘HTML Viewer frame layout’ and complete the initial display of the viewer.

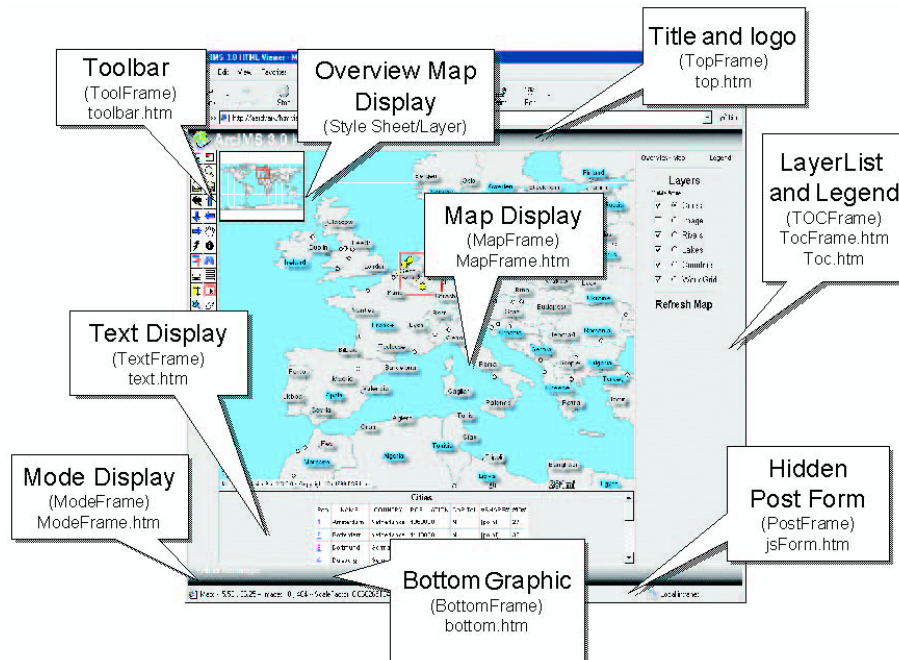
Files used to load and display data

The following files either load a form or display data when called.

- buffer.htm
- query.htm
- storedquery.htm

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. Viewer.htm defines the frame layout of the HTML Viewer created by ArcIMS Designer and is shown below.



TopFrame

Top.htm defines the content for TopFrame. TopFrame is across the top of the viewer and displays the title text (set in viewer.htm) 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 many of the JavaScript files that 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 stylesheet in Internet Explorer or a layer in Netscape. The Overview Map tool on the toolbar toggles the visibility.

The scalebar 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 pointer 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

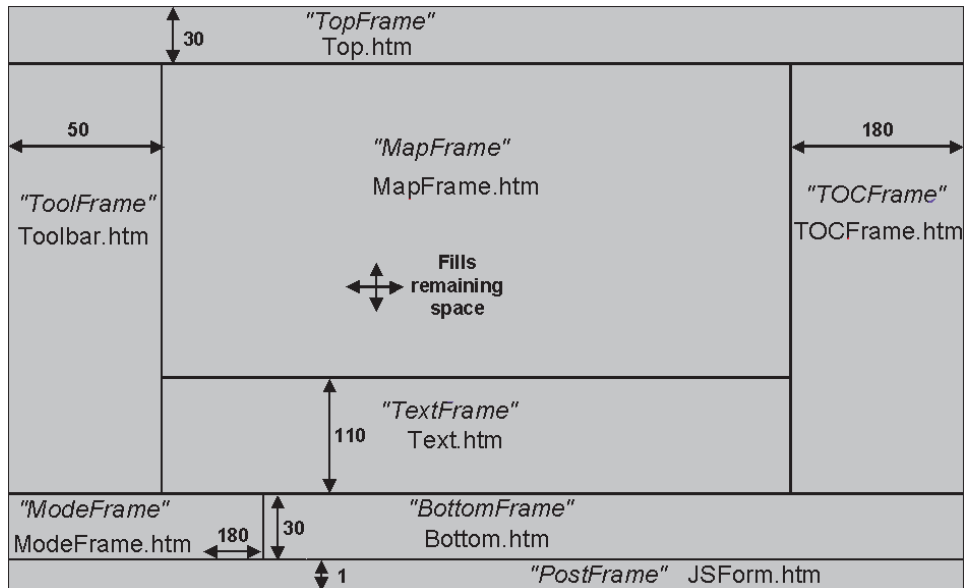
JsForm.htm defines the content for PostFrame. This form is used for communication with the ArcIMS Servlet Connector. PostFrame is hidden and contains the hidden form named Post Form. See ‘How the viewer and server communicate’ in Chapter 2, ‘Customizing the HTML Viewer’, for more information on communication with the ArcIMS Servlet Connector.

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.

Multiple hosts

Typically, Web sites created by Designer access only the services set up on the same Web server where the Web site pages reside. In other words, the URL defining the Web site contains the same Web server host as the URL defining the service. For example, if Web pages reside in the \\my_computer\\Website directory, then typically the URL to the viewer is *http://my_computer/Website/my_Website*, and the URL to the services is *http://my_computer/servlet/com.esri.esrimap.Esrimap?Service=my_service*. The variable defining the service is named `imsURL` and found in `ArcIMSParam.js` file.

When you want to run your ArcIMS services on a server different from the server hosting your HTML Viewer (your Web site pages viewer), you must do one of the following to avoid violating the JavaScript security limitation, which allows for access to only one host:

- Perform redirection tasks—These tasks let the ArcIMS Application Server know it's okay to use multiple hosts and tell it which hosts, by name, are allowed; requests are then forwarded, or “redirected”, to the appropriate host. You can specify that all hosts be allowed to receive redirected requests. When you use redirection, you cannot use authentication, that is, you cannot restrict access.
- Use the Java Connector—Using this connector lets you avoid performing redirection tasks but requires you to install the Java Connector. You can use authentication with the Java Connector, but you must develop the code for it yourself (no code sample is provided for authentication in the HTML Viewer or the Java Connector).

Performing redirection tasks

You can access services on another Web server by modifying properties of the ArcIMS default servlet connector, which can redirect the request to the servlet connector on the other Web server. The response from the connector on the other Web server is packaged in the form page (`jsForm.htm`) and sent back to the viewer. This allows the Web pages to communicate only with the original host and to behave as if it were handled locally. For more information on requests that are sent between the viewer and server, see the section in Chapter 2, ‘How the viewer and server communicate’.

Redirection for the HTML Viewer requires changing properties of the ArcIMS default servlet connector. Search for the file `Esrimap_prop` (the location depends on your Web server or servlet engine) and add the following two lines, replacing `host1`, `host2`, and so on, with valid host names:

```
redirect=true
redirectableHosts=host1,host2,host3,host4
```

Save the file, then stop and start your servlet engine.

The first line tells the Servlet Connector that redirection is allowed. The second line lists the hosts to which redirection is allowed.

An error occurs if redirection is requested to a host other than those listed under `redirectableHosts`. To allow redirection to all hosts, use:

```
redirectableHosts=*
```

Using the Java Connector

While the main purpose of the Java Connector is to enable custom applications built using Java or JavaServer Pages (JSP) to work with ArcIMS services, you can also use it with the HTML Viewer:

- To avoid having to perform redirection tasks in multiple-host configurations. The Java Connector does not violate the JavaScript security limitation, which limits access to one host only.

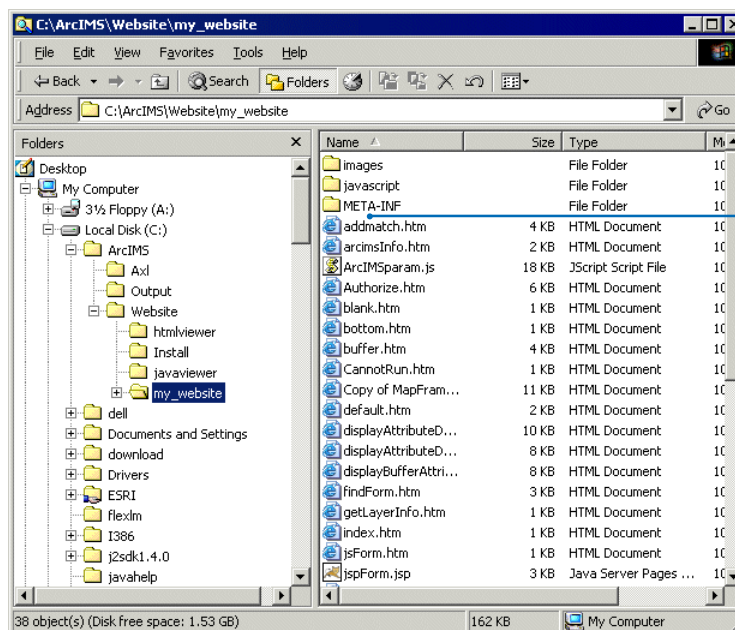
- To prevent end users with ArcMap from accessing your source data. An end user with ESRI's ArcMap application can download the source data used in an ArcIMS service. To prevent ArcMap users from accessing this data, you can use the Java Connector with the HTML Viewer. Using the Java Connector (or any other connector) allows you to remove the ArcIMS Servlet Connector from the production Web server. Without direct access to the ArcIMS Servlet Connector, the ArcMap user cannot access the source data. If you use the Java Connector and remove the ArcIMS Servlet Connector, all clients that use the ArcIMS Servlet Connector, including ArcMap, ArcPad®, and ArcExplorer™ Java, will no longer be able to access ArcIMS services.

Regardless of which reason you are using the Java Connector, you perform the same setup and deployment procedures. If you wish to use authentication with this configuration option, you must develop the code for it yourself, in Java or JSP.

By default, the HTML Viewer communicates through the ArcIMS default servlet connector using HTML forms (viewer.htm file points to jsForm.htm).

Setting up the Java Connector with the HTML Viewer

- If you haven't already, create an HTML Viewer using Designer.
- Using the following steps, copy an ArcIMS—Java Connector WEB-INF directory to the top level of the HTML Viewer you created.
 - Use your operating system's standard controls to navigate to an ArcIMS Java Connector WEB-INF directory. You can use the WEB-INF directory for the ArcIMS Service Administrator, or you can install the Java and JSP sample applications using the ArcIMS installation wizard and use one of the WEB-INF directories in the sample applications. The default directory to which the ArcIMS Service Administrator WEB-INF directory is installed is <ArcIMS Installation Directory>\ArcIMS\Administrator\esriadmin.
 - Use your operating system's standard controls to copy the WEB-INF folder you've located.
 - Use your operating system's standard controls to navigate to the top level of the HTML Viewer you created, then paste the WEB-INF directory there. An illustration showing the location in Windows 2000 follows.



Paste the WEB-INF folder to this location.

3. In the top level of the HTML Viewer you created, navigate to the viewer.htm file.
4. Open viewer.htm in a text editor.
5. Find the var connectorType variable.
6. Change its value from `Servlet` to `JSP` as shown in the following sample.

```
var connectorType = "JSP";
```

7. Save and close viewer.htm.
8. Navigate to the ArcIMS Installation Guide. By default, it is installed at <ArcIMS Installation Directory>\ArcIMS\Documentation\Install.htm.

9. Deploy your HTML Viewer:

Open step 4 of the ArcIMS Installation Guide and navigate to the section, 'Configuring your Web server for ArcIMS service administration'. Follow the instructions in this section. The process for deploying your HTML Viewer is the same as it is for deploying your ArcIMS Service Administrator, except you must substitute the location and directory name of the HTML Viewer you created for the location and directory name of the Service Administrator.

10. Restart the servlet engine.

Customizing the HTML Viewer

2

IN THIS CHAPTER

- **Working with ArcIMSParam.js**
- **Working with services 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 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 ArcIMSParam.js

When you are ready to create a custom Web site, the ArcIMSParam.js file is likely the first file you'll modify. ArcIMSParam.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, 'HTML Viewer JavaScript Library'.

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

- Changing services 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
- Returning unique values only
- Allowing queries
- Displaying a second scalebar

The following pages show you how to modify the ArcIMSParam.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 ArcIMSParam.js unless otherwise noted.

CheckParams function

ArcIMSParam.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 simply 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 become incorporated in the Web site.

Returning unique values only

You can modify ArcIMSParam.js to return only unique values in the Query form's Get Sample Values.

1. In ArcIMSParam.js, set the value of onlyUniqueSamples to "true". You may have to increase the number of sample values returned by setting the value of numberDataSamples to a higher number. This value is the total number of all returned values, not the number of unique values.

```
var onlyUniqueSamples = true;
// queries are case insensitive?

// number of data samples retrieved for query form
var numberDataSamples = 50;
```

Allowing queries to not be case-sensitive

You can enable case-insensitive queries. By default, queries are case-sensitive to behave the same way they do in the ArcIMS standard and custom Java Viewers and in ArcExplorer.

1. In ArcIMSParam.js, set the value of queryCaseInsensitive to true.

```
var queryCaseInsensitive=true;
```

Displaying a second scalebar

You can display a second scalebar. By default, the first ScaleBar units are set to miles, while the second ScaleBar units are set to kilometers. The units of the second scalebar cannot be modified by editing the ArcIMSParam.js file. The units of the second scalebar are dependent on the units of the first scalebar. This dependency, which cannot be changed, works as follows:

If the units of the first scalebar are set to miles, then the viewer sets the units of the second scalebar to kilometers. If the first is set to kilometers, then the second is set to miles. If the first is feet, then the second is meters. If the first is meters, then the second is feet.

To display a second scalebar

1. In ArcIMSParam.js, set the value of drawScaleBar2 to true as shown in the following sample.

```
var drawScaleBar2 = true;
```

2. Set the ScaleBar2 parameters as needed, as shown in the following sample.

```
var ScaleBar2Units = "KILOMETERS";
var ScaleBar2Background = "false";
var ScaleBar2BackColor = "0,0,0";
var ScaleBar2FontColor = "0,0,0";
var ScaleBar2Color = "128,128,128";
var ScaleBar2Font = "";
var ScaleBar2Style = "Regular";
var ScaleBar2Round = "1";
var ScaleBar2Size = "9";
var ScaleBar2Width = "5";
var ScaleBar2Precision = 2;
```

Working with services and map layout

Changing the service

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

Changing the overview service

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

Removing the overview map

Set the variable *hasOVMap* to false in *ArcIMSParams.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.

Moving the overview map into a separate frame

In the default HTML Viewer, the overview map displays on top of the main map in *MapFrame*. You can specify that the overview map display in its own frame by performing the following procedure.

1. Add a frame to *viewer.htm* called *OverviewFrame* that loads *overview.htm*. Copy *overview.htm* from the Thematic sample directory, for example, *C:\ArcIMS\Website\htmlviewer\Thematic*, to the top level of your viewer directory.
2. In *MapFrame.htm*, modify the lines of the Cascading Style Sheets (CSS) layers referencing the overview map and extent box to match the sample code below. If you have not yet customized the *MapFrame.htm* file, the bold text in the sample code signifies the exact changes you must make.

```
// overview map and shadow
content = '';
createLayer("ovShadow",-10,-10,1,1,false,content);
//if ((isNav4) || (isIE)) clipLayer("ovShadow",0,0,1,1);
content = '';
createLayer("ovLayer",-10,-10,1,1,false,content);
setLayerBackgroundColor("ovLayer", "white");
// overview extent box
content = '';
createLayer("zoomOVBoxTop",-10,-10,1,1,false,content);
content = '';
createLayer("zoomOVBoxLeft",-10,-10,1,1,false,content);
content = '';
createLayer("zoomOVBoxRight",-10,-10,1,1,false,content);
content = '';
createLayer("zoomOVBoxBottom",-10,-10,1,1,false,content);
```

The CSS layers are not removed (removing them causes errors) but are instead resized to one pixel in width and height.

3. In MapFrame.htm set the variable `ovIsVisible` to true as shown in the following sample code.

```
ovIsVisible = true;
```

4. In ArcIMSParam.js, set the variable `ovMapIsLayer` to false as shown in the following sample code.

```
// does the overview map a layer on top of map?...
var ovMapIsLayer=false;
```

This informs the viewer that the overview is not overlaying the main map display (that it won't be present in MapFrame) and to not account for the coordinates within the area occupied by the overview map while it is visible.

5. Create a new frame for the overview map. Open viewer.htm and add the following bold lines of code.

```
<FRAMESET ROWS="*,110">
  <FRAME NAME="MapFrame" SRC="MapFrame.htm" MARGINWIDTH="0" MARGINHEIGHT="0"
  SCROLLING="No" FRAMEBORDER="Yes" RESIZE="YES">
  <FRAME NAME="TextFrame" SRC="text.htm" MARGINWIDTH="0" MARGINHEIGHT="0"
  SCROLLING="Auto" FRAMEBORDER="Yes" RESIZE="Yes">
</FRAMESET>
document.writeln('<FRAMESET ROWS="*,150">');
document.writeln('<FRAME NAME="TOCFrame" SRC="TOCFrame.htm" MARGINWIDTH="0"
MARGINHEIGHT="0" SCROLLING="Auto" FRAMEBORDER="Yes">');
document.writeln('<FRAME NAME="OverviewFrame" SRC="overview.htm" MARGINWIDTH="0"
MARGINHEIGHT="0" SCROLLING="No" FRAMEBORDER="Yes">');
document.writeln('</FRAMESET>');
</FRAMESET>
<FRAMESET COLS="180,*" FRAMEBORDER="0" BORDER=0 FRAMESPACING="0">
```

6. Locate the function `checkParams()` and change the value of the `ovImageVar` variable. This variable references the image object that displays in the overview map. Change this variable's value from the image object in MapFrame.htm to the image object in the frame OverviewFrame.htm, as shown in the following sample code.

```
// global for overview map. . . change if not on same frame as Map
ovImageVar = parent.OverviewFrame.document.ovImage;
```

7. Save your work and view your site. The overview map should be in its own frame.

Changing the starting map extent

Set the values of these four variables to the new coordinates: *startLeft*, *startBottom*, *startRight*, and *startTop*. This sets the starting x,y coordinates of the map extent.

Displaying the graphic legend at startup instead of the LayerList

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

In ArcIMSParam.js, set the variable `showTOC` to false, add the variable *legendVisible*, and set it to true in the function `checkParams` as shown in the following sample.

```
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 `ArcIMSParam.js` file set the variable *ActiveLayerIndex* to the index number of the layer you want active. An index number of 0 indicates the first (top) 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. In this example, the `Customers` layer is being set as the active layer. In the map configuration file, the `<LAYER>` elements 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 splashes
content = '';
createLayer("LoadData",loadBannerLeft,loadBannerTop,273,30,false,content);
content = '';
createLayer("LoadMap",loadBannerLeft,loadBannerTop,273,30,false,content);
```

Immediately following the `<BODY>` element, 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);
```

Passing map parameters in the startup URL

Several parameters can be passed to the viewer on startup. The JavaScript function that parses these parameters from the loading URL is found in `aimsMap.js` and is named `getCommandLineParams`. The parameters are put into the URL for the viewer in the following format:

```
http://fullSitePath/
default.htm?Parameter1=Value1&Parameter2=Value2&Parameter3=Value3
```

The available parameters that can be passed are:

<code>Service=MapService</code>	Name of service. Replaces one defined in <code>ArcIMSParam.js</code> .
<code>OVMMap=OvMapService</code>	Name of service for overview map. Replaces one defined in <code>ArcIMSParam.js</code> .
<code>Box=minX:minY:maxX:maxY</code>	Extent to be displayed. Coordinates are separated by colons.
<code>MaxRect=minX:minY:maxX:maxY</code>	Limit of extent viewer can use. Coordinates are separated by colons.
<code>Layers=01011</code>	Visible layers, starting from top layer; 0=invisible; 1=visible. Only affects layers listed in parameter value. Example only affects the top five layers.
<code>ActiveLayer=layerIndex</code>	Index of layer to be set to active layer.
<code>Query=queryExpression</code>	Query expression to be sent. Query expression is executed against active layer and must be URL encoded. (<code>'CITY = "New York"'</code> becomes <code>'CITY%20%3D%20%22New%20York%22'</code> .)
<code>QueryZoom=Yes</code>	Zoom to area around first feature in selection set returned from passed query expression.
<code>Extent=Auto</code>	Force viewer to obtain start and limit extents from server. Replaces start and limit parameters set in <code>ArcIMSParam.js</code> .

Use multiple hosts

When you want to run your ArcIMS services on a server different from the server hosting your HTML Viewer (your Web site pages viewer), you must take measures to avoid violating the JavaScript security limitation, which allows for access to only one host. For more information, see the 'Multiple hosts' section in Chapter 1 of this guide.

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 scalebar. You can change these elements or add new elements.

Controlling layers displayed in Legend and LayerList

To prevent certain layers from being listed in the LayerList or Legend, in `ArcIMSParam.js` set the value of the variable `hideLayersFromList` to true. If this variable is true, then elements in the array `noListLayer` must be set up. An element must be defined for each layer designating if it will be prevented from being listed or not, starting with the top layer as element zero.

In the following example the fourth layer will not be listed in either the LayerList or Legend:

```
// toggle the check of nonlisting of layers in LayerList and Legend
// if true, noListLayer array must have an element defined for each layer
var hideLayersFromList=true
// layers that will not be listed in the LayerList or Legend
// Note: This does not affect map display
var noListLayer = new Array();
  noListLayer[0] = false;
  noListLayer[1] = false;
  noListLayer[2] = false;
  noListLayer[3] = true;    // this one will not be listed
  noListLayer[4] = false;
```

This will not affect the actual display of the layer within the map image.

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 scalebar

The scale bar is displayed if `drawScaleBar` is true. To modify the style, font, size, color, and so on, use the variables `ScaleBarStyle`, `ScaleBarFont`, `ScaleBarFontColor`, `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. Field names must be in uppercase to match what the ArcIMS Spatial Server returns.

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 top layer set at index 0. Each line of the array is assigned as follows:

```
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, 'HTML Viewer JavaScript Library'.

Hiding display of ID and Shape field

The ID and Shape fields are, by default, listed in the displayed attribute data returned from an Identify/Query request. To hide the display of these fields, in *ArcIMSParam.js*, set the variables *hideIDFieldData* and *hideShapeFieldData* to true as shown below:

```
var hideIDFieldData = true;
var hideShapeFieldData = true;
```

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 top 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 as follows:

```
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, 'HTML Viewer JavaScript Library'.

When using the Query Builder tool, the dropdown list and query expression show different field names. The dropdown list displays the *alias* field names as defined in *fieldAliasList*. The query expression, however, is constructed with the *original* field names in order to correctly process the request against the database.

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.

Identifying all visible features at one location

In the default viewer the Identify tool works only on the current Active Layer. The viewer also has an IdentifyAll tool that sends a query request to identify features at one click point for each visible feature layer. In *ArcIMSParam.js*, set the variables *useIdentify* and *useIdentifyAll* as shown below. Only one of these can be true, with priority going to *useIdentify*.

```
var useIdentify = false;  
var useIdentifyAll = true;
```

Examples of the IdentifyAll tool are found in the Hyperlink and Extract samples.

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 Viewer'.

You can change the title after the HTML Viewer is created by editing the viewer.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 upper 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, name of the image, or both 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, name of the image, or both to change the background appearance of the topFrame. You should also change this graphic in ModeFrame.htm and bottom.htm to match.

If your logo varies greatly in size from the original, you will want to edit the width of the frame in viewer.htm. Edit the following line by changing 30 to a larger number that accommodates the width of the logo graphic. The first 30 is for the top frame, and the second is for the bottom frame.

```
document.writeln('<FRAMESET ROWS="' + (30+addNS) + ',*,30,0" FRAMEBORDER="No"
FRAMESPACING="0" onload="doIt()" BORDER=0 ' + moreStuff + '>');
```

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

Changing search tolerance for Identify and Hyperlink

Set the value of *pixelTolerance* to change the number of pixels used for the search tolerance during an Identify or Hyperlink 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;
```

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.

```
var usePan=true;
var usePanNorth=false;
var usePanWest=false;
var usePanEast=false;
var usePanSouth=false;
```

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, refer to the Extract sample application. 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, 'HTML Viewer JavaScript Library'. You can also look at the 'Using the sample HTML Viewers' section later in this chapter.

The value set for the search tolerance around the click point can affect the user's ability to find the hyperlinked point. Modify the value of *pixelTolerance* to change the number of pixels used for the search tolerance for a hyperlink. The default is 2.

```
var pixelTolerance=5;
```

Linking to the first visible feature hyperlink on a map

In the default viewer the Hyperlink tool works only on the current Active Layer. The viewer also has a *useHyperlinkAny* tool that sends a query request to link the first visible feature with defined hyperlinks at one click point.

In *ArcIMSParam.js*, set the variables *useHyperlink* and *useHyperlinkAny* as shown below. Only one of these can be true, with priority going to *useHyperlink*.

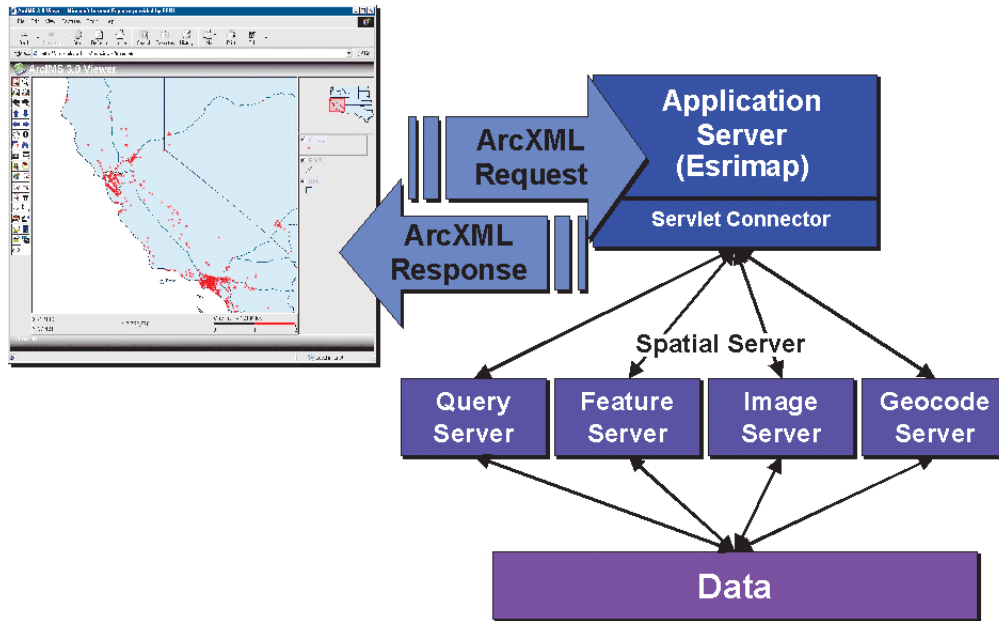
```
var useHyperlink = false;
var useHyperlinkAny = true;
```

An example of the *useHyperlinkAny* tool is found in the Hyperlink sample.

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 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 function type (and to the correct host server, if different servers are used for services and the viewer), then returns the response to the viewer.



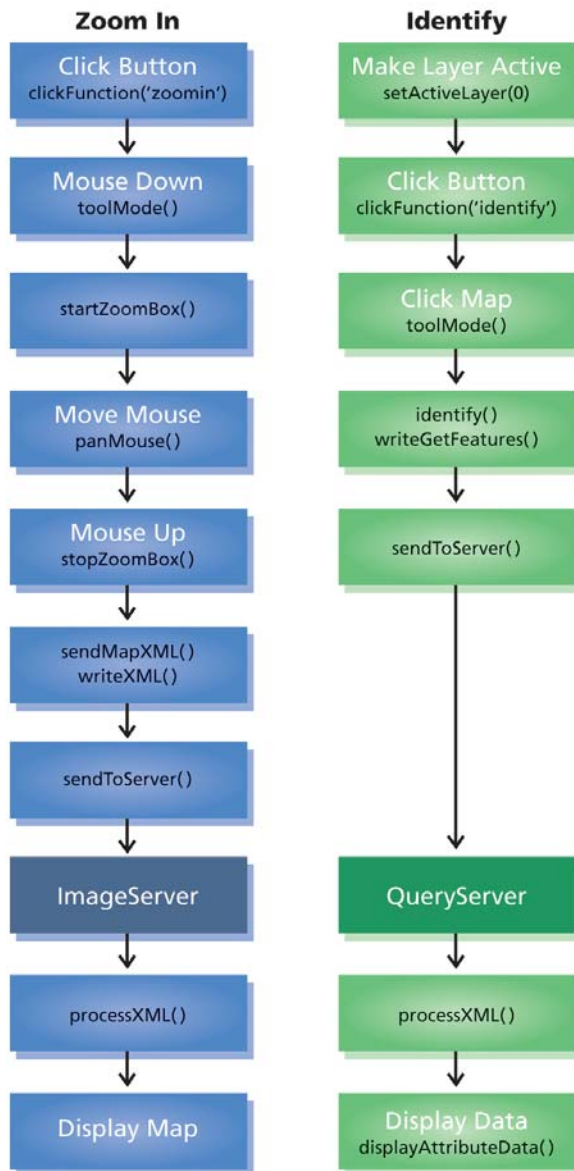
For an explanation and diagram of the entire ArcIMS architecture, including the Application Server and Spatial Servers, see the architecture topics in the ArcIMS online Help system.

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—what happens when the mouse is clicked (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



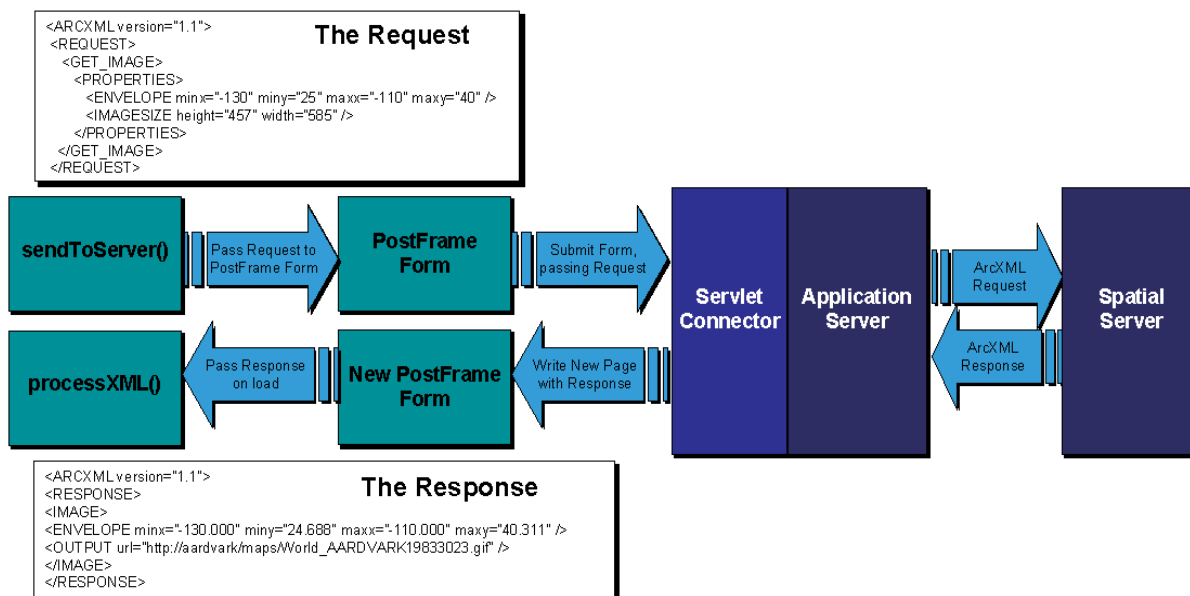
The ArcXML request and response cycle

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 attributes and input elements of the PostFrame form are updated using JavaScript. They are passed to the Servlet Connector by submitting the form. A new page is written with the response and 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. 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 and is submitted to the Web server. The Servlet Connector extracts the submitted values and sends the request on to the Application Server. The Application Server then sends the request to the ArcIMS Spatial Server.



The response

The ArcXML response from the ArcIMS Spatial Server is sent back to the Servlet Connector through the Application Server. A new HTML page is dynamically written back to the PostFrame and replaces the previous HTML 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 (the post form) works

The PostFrame originates as jsForm.htm and is replaced during the response. Below is the original jsForm.htm file.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<HTML>
<HEAD>
  <TITLE>Default Form</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="ArcXMLRequest" VALUE="">
  <INPUT TYPE="Hidden" NAME="JavaScriptFunction"
VALUE="parent.MapFrame.processXML">
  <INPUT TYPE="Hidden" NAME="RedirectURL" VALUE="">
  <INPUT TYPE="Hidden" NAME="BgColor" VALUE="#000000">
  <INPUT TYPE="Hidden" NAME="FormCharset" VALUE="UTF-8">
</FORM>
</BODY>
</HTML>
```

When a request is sent to the ArcIMS Spatial Server, the input elements of the PostFrame form are updated with the following:

ACTION: URL of service

ArcXMLRequest: ArcXML request

JavaScriptFunction: The function that will process the response. Default is parent.MapFrame.processXML.

BgColor: Color of page background. Default is black.

FormCharset: Character set encoding of ArcXML. Default is UTF-8.

The values are updated, and the form is submitted to the Servlet Connector. The Servlet Connector extracts the ArcXML request and forwards it through the Application Server to the appropriate part of the ArcIMS Spatial Server for processing.

During the response from the ArcIMS Spatial Server, the contents of the PostFrame are replaced. Below is an example of the new PostFrame file.

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=UTF-8">
<HTML>
<HEAD>
<TITLE>Default Form</TITLE>
<!-- Title must match jsForm.htm's title -->
<SCRIPT TYPE="text/javascript" LANGUAGE="JavaScript">
function passXML()(
  var XMLResponse='<?xml version="1.0" encoding="UTF8"?><ARCXML version="1.1">
  <RESPONSE><IMAGE><ENVELOPEminx="-180" miny="-126" maxx="180" maxy="126"/>
  <OUTPUT file="c:\\arcims\\output\\World_KAT27721614.gif" url="http://kat/
  output/World_KAT27721614.gif"/></IMAGE></RESPONSE></ARCXML>';
  parent.MapFrame.processXML(XMLResponse);
```

```

}
</SCRIPT>
</HEAD>
<BODY BGCOLOR="#000000" 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="BgColor" VALUE="#000000">
<INPUT TYPE="Hidden" NAME="FormCharset" VALUE="UTF-8">
<INPUT TYPE="Hidden" NAME="RedirectURL" VALUE=" ">
</FORM>
</BODY>
</HTML>

```

When this page has finished loading, the ArcXML response is passed to the JavaScript function specified in the form element JavaScriptFunction. This function then processes the response.

The GET_SERVICE_INFO request

The GET_SERVICE_INFO request is one of the more commonly used requests because it requests information about each layer in an ArcIMS service. It is helpful to have an understanding of this request, especially when trying to understand how the overview map works, how to load layer parameters from file, and how map units work.

With Image and Feature Services, this request has options for returning information on the fields, envelope, extensions, and renderers. With ArcMap Image Services, the request has options for returning information on fields, envelope, data frames, and the table of contents.

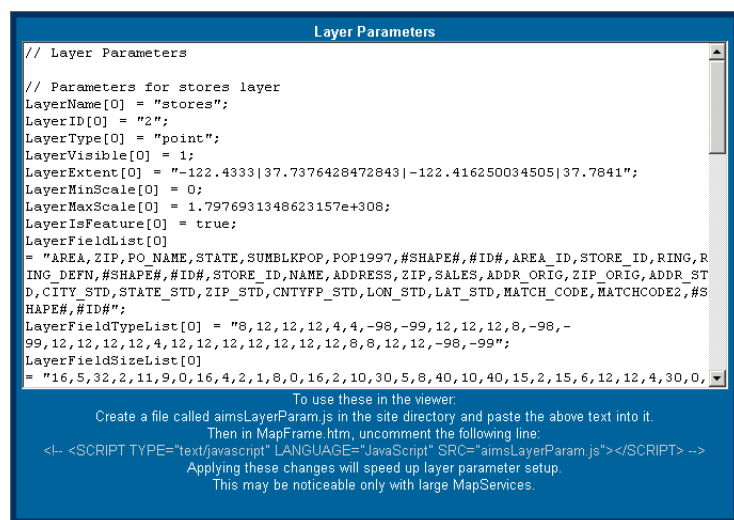
Load layer parameters from file

By default, the HTML Viewer uses a single GET_SERVICE_INFO request or response for all basic information. But you may be able to decrease viewer startup time by specifying that you want to load the layer parameters from file rather than by parsing them from the GET_SERVICE_INFO response. Loading from file decreases viewer startup time when loading services with many layers, services, or both layers and services containing many attribute fields.

1. If you haven't already, create an HTML Viewer using Designer.
2. Open a Web browser and type the following in the address box.
`http://<your host name>/<your Web site directory>/<your Web site name>/getLayerInfo.htm`

The Layer Parameters box (illustration to the right) displays in your browser.

3. Open a text editor, then copy the code in the main area of this box and paste it into the text editor. Save the pasted code as a file with a filename of aimsLayerParam.js.



The above box appears after you perform step 2. It provides instructions and code you copy and paste into a new file to help you load parameters from file.

4. Save the file you just created in the javascript subfolder of your Web site directory.
5. In a text editor, open MapFrame.htm, located at the top level of your Web site directory.
6. Find the following line of code and uncomment it to activate it.

```
<SCRIPT TYPE="text/javascript" LANGUAGE="javascript" SRC="javascript/
aimsLayerParam.js"></SCRIPT>
```

To uncomment code, delete the beginning comment tag `<!--` and the ending comment tag `-->`, then save the file.

The overview map

The viewer, for the overview map, makes a GET_IMAGE request without an extent envelope. The Spatial Server returns an image with the extent defined in the overview map's configuration file (.axl file). Since the extent envelope is not sent in the request, there is no need to determine the decimal delimiter type (point or comma) used by the Spatial Server. The overview map ignores the limit extent coordinates defined in the ArcIMSParam.js file.

Map units

The viewer gets its map units from the GET_SERVICE_INFO response. The global variable MapUnits holds this value. This is essential for loading services when the map units are not known, such as in the Generic, JavaPost, and MultiService samples, or in a custom implementation.

JavaScript function files

The following JavaScript function files are included in the HTML Viewer. The ArcIMSParam.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, 'HTML Viewer JavaScript Library'.

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

Using the sample HTML Viewers

Several sample implementations of the HTML Viewer are provided with ArcIMS. They demonstrate a variety of functions and graphic user interface (GUI) designs. You should have a working knowledge of creating services and browsing ArcIMS Web sites to work with the samples.

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

The samples are Basic Map, Extract, Generic Map, HyperLink, Java Post, MultiService, Parcels, and Thematic Map.

A description and requirements for running each sample are provided below. You can also reference the setup instructions in your \<installationdirectory>\Samples\Viewers\HTMLSample_setup.htm file. These descriptions assume you used the typical installation option as defined in 'General instructions for setting up samples' in this same file. If you did a custom installation, reference this topic for complete instructions.

Basic Map

Description:

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

Setup requirements:

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

Web site files:

In \htmlviewer\BasicMap, most of the functions are defined in the MapFrame.htm and Toolbar.htm files. The MapFrame filenames 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 the following four files from the JavaScript Function Files list above.

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

Extract

Description:

This sample viewer demonstrates the use of Extract Server to extract layers into a shapefile based on a user-defined extent.

Setup requirements:

1. Create an Image Service named sanfrancisco from sf.axl.
2. In the browser, type in the URL to your host Website htmlviewer directory (for example, `http://<ArcIMS host>/Website/htmlviewer`).
3. Click Extract, and the sanfrancisco service is displayed. To test the extract function, make a layer active and use Select by Rectangle to select features. Click the Extract tool, click Extract, then click Download and choose a location for the ZIP file.

Web site files:

In Viewer.htm notice the overview map is in a frame separate from the main map, and there is a frameset within a frameset. Functions to support a drill-down identify tool and to buffer around a user-defined shape, using the Select by Shape tool, are demonstrated.

The Extract sample uses the following JavaScript files:

- All the files in the JavaScript Function Files list except aimsBuffer.js, aimsGeocode.js, and aimsPrint.js.
- AimsExtract.js—functions that create the dropdown list for the options.
- AimsExtractResource.js—text strings for interface.

Generic Map

Description:

This sample viewer presents a dropdown list of all Image Services running on a server and allows the user to choose one. 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.

Setup requirements:

1. No specific service needs be defined in ArcIMSParam.js, but you want to have at least one Image Service running.
2. In the browser, type in the URL to your host Website htmlviewer directory (for example, `http://<ArcIMS host>/Website/htmlviewer`).
3. Click Generic Map. Choose a service for the main map and overview map, if desired, then click Load. Try out the Options button at the bottom of the toolbar.

Web site files:

In \htmlviewer\Generic, there is an HTML page that defines each option, with a name 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 the following JavaScript files:

- All the files in the JavaScript Function Files list.
- AimsOptions.js—functions that create the dropdown list for the options.
- AimsGeneric.js—functions that create the dropdown lists for loading the requested services.
- AimsGenericResource.js—text strings for interface.

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 Basic Map (described above).

Setup requirements:

1. Create an Image Service named sanfrancisco from sf.axl.
2. In the browser, type in the URL to your host Website htmlviewer directory (for example, `http://<ArcIMS host>/Website/htmlviewer`).
3. Click HyperLink, and a map of cultural features is displayed. Click Queries on the sidebar, click HyperLink, then click an art gallery to link to its home page. Note: Not all art galleries have valid links.

Web site files:

In `\htmlviewer\Hyperlink`, the black custom interface is defined in `viewer.htm`.

The HyperLink sample uses all the files in the JavaScript Function Files list above. Also look at `ArcIMSparam.js` for the parameters `UseHyperLink`, `hyperLinkLayers`, and `hyperLinkFields`.

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 to the ArcIMS Servlet Connector. The applet uses Java 1.1 instead of Java 2 and, therefore, does not require the Java Runtime Environment (JRE) Plug-in. The sample supports the same functions as the HTML Viewer created by Designer but can be extended with Java 1.1.

Setup requirements:

1. Create an Image Service named sanfrancisco from sf.axl.
2. In the browser, type in the URL to your host Website htmlviewer directory (for example, `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. `MapFrame.htm` contains the `sendToServer` function. The `sendToServer` 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.

The Java Post sample uses all the files in the JavaScript Function Files list.

MultiService

Description:

This sample viewer demonstrates the ability to load multiple services. It also shows you how to set up the HTML Viewer to access services from more than one host.

Setup requirements:

1. Create an Image Service named `basicworld` from `BasicWorld.axl`, setting the image type to either GIF or PNG8.
2. While most samples only show services on your local host, this sample allows you to choose services from other hosts. If you wish to do this, you must complete steps 3 and 4. If not, continue to step 5.
3. Search for the file `Esrimap_prop`; its location depends on your Web server type. Open the file and add the following lines, replacing `<hostname#>` with the name of the host you want to get services from.

This is known as redirection because you are redirecting the URL path to another host:

```
redirect=true
redirectableHosts=<hostname1>,<hostname2>...
```

4. In `\htmlviewer\multiservice` find and open `aimsMultiServiceParam.js` and update the variable `availableHostsList` with the same host names you included in `Esrimap_prop` above.
Note that `aimsMultiServiceParam.js` does not support use of the wildcard character (*) to signify all hosts but instead requires that you list each host name. This behavior differs from `Esrimap_prop`, which allows you to use the wildcard character to indicate all hosts.
5. In the browser, type in the URL to your host Website `htmlviewer` directory (for example, `http://<ArcIMS host>\Website\htmlviewer`).
6. Click **MultiService** and try adding more than one service using the **Add service** button in the lower right corner. Try selecting a second host if you have set your properties to do so.

Web site files:

In this sample, many services can be put into one viewer. The image output file type for the services must be GIF or PNG8 because a transparent background is necessary to show one service displayed underneath another service. In `aimsMultiServiceParam.js`, the variable `topMapServiceURL` is used to define the top Image Service. This is also used to display the North arrow, scalebar, and copyright. You must make sure that the service defined here creates GIF or PNG8 image type. If not, the services beneath will be obscured. Internet Explorer and Netscape 6 support transparency for both formats; however, Netscape 4.x does not support transparency for PNG. No projection of services is performed in this sample, so your services must be in the same coordinate space to appear correctly, or you could add your own projection code.

The **MultiService** sample uses all the files in the JavaScript Function Files list except `aimsBuffer.js`, `aimsGeocode.js`, and `aimsPrint.js`.

Parcels

Description:

This sample viewer demonstrates linking 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 uses either Active Server Pages (ASP) or ColdFusion.

Setup requirements:

- For ASP:
 1. An Open Database Connectivity (ODBC) connection to external data is required. Go to Start menu|Settings|Control Panel and choose ODBC Data Sources. Click the System DSN tab, click Add, and select the Microsoft Access Driver (.mdb). Set the Data Source Name to Downtown, and choose downtown.mdb in the <ArcIMS Installation Directory>\Samples\Viewers\Data\downtown directory for the database.
 2. In the \htmlviewer\Parcels directory, copy asp.htm to default.htm.
 3. Proceed to *For all implementations*.
- For ColdFusion:
 1. If not already running, start the ColdFusion Server 4.5.
 2. An ODBC connection to external data is required. Go to Start menu|Settings|Control Panel and choose ODBC Data Sources. Click the System DSN tab, click Add, and choose the Microsoft Access Driver (.mdb). Set the Data Source Name to Downtown, and choose downtown.mdb in the <ArcIMS Installation Directory>\Samples\Viewers\Data\downtown directory for the database.
 3. In the \htmlviewer\Parcels directory, copy cf.htm to default.htm.
 4. Proceed to *For all implementations*.
- For all implementations:
 1. Create an Image Service named parcels from parcels.axl.
 2. In the browser, type in the URL to your host Website htmlviewer directory (for example, <http://<ArcIMS host>\Website\htmlviewer>).
 3. Click Parcels; click Zoom in; and try out 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 a .asp extension.

The Parcels sample uses all the files in the JavaScript Function Files list above, 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.
- AimsDBResource.js—text strings for interface.

Thematic Map

Description:

This sample viewer shows thematic rendering of layers and generating statistics on a field.

Setup requirements:

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

Web site files:

The Thematic sample uses the following JavaScript files:

- All the files in the JavaScript Function Files list except `aimsBuffer.js`, `aimsPrint.js`, and `aimsSelect.js`.
- `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.
- `AimsClassRenderResource.js`—text strings for interface.

HTML Viewer JavaScript Library

3

IN THIS CHAPTER

- **Organization of the HTML Viewer JavaScript Library**
- **JavaScript functions by name**
- **JavaScript functions by category**
- **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.

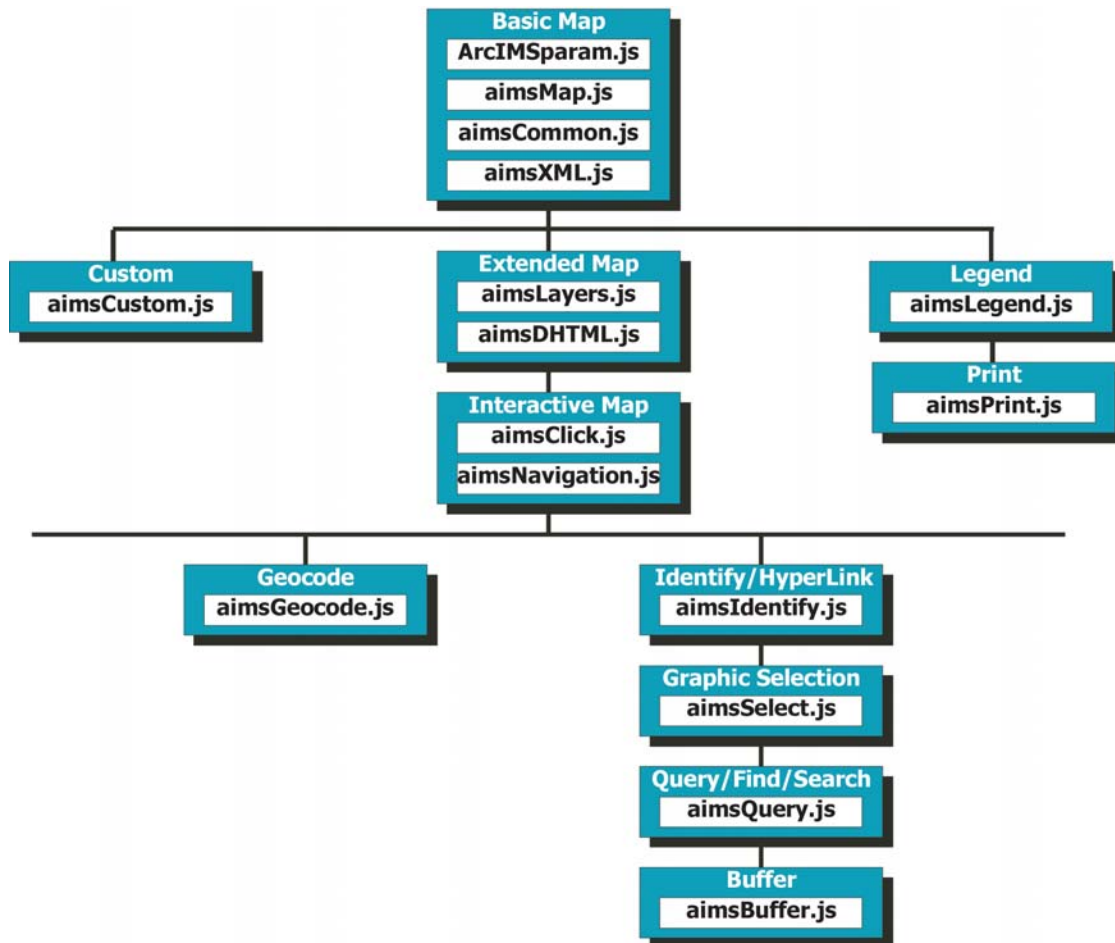
This chapter provides an overview and references to the JavaScript functions and global variables contained in the HTML Viewer.

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 is 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: ArcIMSParam.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.

ArcIMSParam.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.

Identify/HyperLink

The Identify/HyperLink category consists of a single file: `aimsIdentify.js`. Use of this file requires the files from the Basic Map, Extended Map, and Interactive Map categories.

`AimsIdentify.js` contains functions to add basic query capabilities to the viewer. Identify 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 Identify/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, Identify/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, Identify/HyperLink, Select, and Query categories.

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

JavaScript functions by name

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	Graphic Selection
afterMapRefresh	aimsMap.js	Basic Map
beforeMapRefresh	aimsMap.js	Basic Map
boxIt	aimsDHTML.js	Extended Map
bufferIt	aimsBuffer.js	Buffer
calcDistance	aimsMap.js	Basic Map
calcSelectEnvelope	aimsSelect.js	Graphic Selection
checkCoords	aimsCommon.js	Basic Map
checkCurrentExtent	aimsCommon.js	Basic Map
checkForForbiddenTags	aimsCommon.js	Basic Map
checkFullExtent	aimsMap.js	Basic Map
checkHyperLinkLayer	aimsIdentify.js	Identify/HyperLink
checkIfActiveLayerAvailable	aimsIdentify.js	Identify/HyperLink
checkParams	ArcIMSparam.js	Basic Map
checkSelected	aimsIdentify.js	Identify/HyperLink
checkStoredQueries	aimsQuery.js	Query/Find/Search
chkMouseUp	aimsClick.js	Interactive Map
clearError	aimsCommon.js	Basic Map
clearLeadingSpace	aimsCommon.js	Basic Map
clearSelection	aimsSelect.js	Graphic Selection
clickAddPoint	aimsClick.js	Interactive Map
clickFunction	aimsClick.js	Interactive Map
clipLayer	aimsDHTML.js	Extended Map
compare	aimsQuery.js	Query/Find/Search
convertDecimal	aimsCommon.js	Basic 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	aimsIdentify.js	Identify/HyperLink
displayAttributeDataforDrill	aimsIdentify.js	Identify/HyperLink
doidentifyAll	aimsIdentify.js	Identify/HyperLink

JavaScript functions by name (continued)

Function Name	File	Category
extractIt	aimsCustom.js	Custom
findForm	aimsQuery.js	Query/Find/Search
formatDate	aimsCommon.js	Basic Map
fixSingleQuotes	aimsCommon.js	Basic Map
fullExtent	aimsMap.js	Basic Map
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/Find/Search
getGeocodeLayers	aimsGeocode.js	Geocode
getGeocodeParams	aimsGeocode.js	Geocode
getHost	aimsXML.js	Basic Map
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	Graphic Selection
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
getService	aimsXML.js	Basic Map

JavaScript functions by name (continued)

Function Name	File	Category
getStartExtent	aimsCommon.js	Basic Map
getStoredQueries	aimsQuery.js	Query/Find/Search
getXMLErrorMessage	aimsXML.js	Basic Map
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
htmlSendToServer	aimsXML.js	Basic Map
hyperLink	aimsIdentify.js	Identify/HyperLink
hyperLinkAny	aimsIdentify.js	Identify/HyperLink
Identify	aimsIdentify.js	Identify/HyperLink
IdentifyAll	aimsIdentify.js	Identify/HyperLink
isNotSameHostInURL	aimsXML.js	Basic Map
isVisible	aimsDHTML.js	Extended Map
jspSendToServer	aimsXML.js	Basic 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/Find/Search
parseFieldSamplesUnique	aimsQuery.js	Query/Find/Search
parseGeocodeLayers	aimsGeocode.js	Geocode
parseGeocodeParams	aimsGeocode.js	Geocode
parseGeocodeResults	aimsGeocode.js	Geocode
parseHyperLink	aimsIdentify.js	Identify/HyperLink
parseHyperLinkAny	aimsIdentify.js	Identify/HyperLink
parseRecordString	aimsCommon.js	Basic Map

JavaScript functions by name (continued)

Function Name	File	Category
parseStartQuery	aimsQuery.js	Query/Find/Search
parseStoredQueries	aimsQuery.js	Query/Find/Search
printIt	aimsPrint.js	Print
processStartExtent	aimsCommon.js	Basic Map
processXML	aimsXML.js	Basic Map
putExtentOnOVMap	aimsDHTML.js	Extended Map
queryForm	aimsQuery.js	Query/Find/Search
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	Graphic Selection
sendCustomToServer	aimsXML.js	Basic Map
sendMapXML	aimsXML.js	Basic Map
sendQueryString	aimsQuery.js	Query/Find/Search
sendShapeSelect	aimsSelect.js	Graphic Selection
sendStoredQuery	aimsQuery.js	Query/Find/Search
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
setStartQuery	aimsQuery.js	Query/Find/Search
setupGeocode	aimsGeocode.js	Geocode
setupLayerVisible	aimsLayers.js	Extended Map
setZoomColor	aimsNavigation.js	Interactive Map
showHighlight	aimsSelect.js	Graphic Selection
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

JavaScript functions by name (continued)

Function Name	File	Category
startSelectBox	aimsSelect.js	Graphic Selection
startUp	aimsCommon.js	Basic Map
startZoomBox	aimsNavigation.js	Interactive Map
startZoomOutBox	aimsNavigation.js	Interactive Map
stopPan	aimsNavigation.js	Interactive Map
stopSelectBox	aimsSelect.js	Graphic Selection
stopZoomBox	aimsNavigation.js	Interactive Map
stopZoomOutBox	aimsNavigation.js	Interactive Map
storedQueryForm	aimsQuery.js	Query/Find/Search
swapQuotes	aimsCommon.js	Basic Map
swapStuff	aimsCommon.js	Basic Map
tempGetSamples	aimsQuery.js	Query/Find/Search
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	Graphic Selection
writeFieldSample	aimsQuery.js	Query/Find/Search
writeFindRequest	aimsQuery.js	Query/Find/Search
writeGeocodeXML	aimsGeocode.js	Geocode
writeGetBufferedData	aimsBuffer.js	Buffer
writeGetFeatures	aimsIdentify.js	Identify/HyperLink
writeGetFeatures2	aimsSelect.js	Graphic Selection
writeGetFeatures3	aimsSelect.js	Graphic Selection
writeGetFeaturesDrill	aimsIdentify.js	Identify/HyperLink
writelnIdentifyXML	aimsIdentify.js	Identify/HyperLink
writeLayerListForm	aimsLayers.js	Extended Map
writeModeFrame	aimsCommon.js	Basic Map
writeModeLayers	aimsCommon.js	Basic Map
writeOVXML	aimsXML.js	Basic Map
writePrintPage	aimsPrint.js	Print
writeQueryBufferXML	aimsBuffer.js	Buffer
writeQueryForm	aimsQuery.js	Query/Find/Search
writeQueryXML	aimsQuery.js	Query/Find/Search
writeShapeBufferXML	aimsBuffer.js	Buffer
writeShapeSelect	aimsSelect.js	Graphic Selection

JavaScript functions by name (continued)

Function Name	File	Category
writeStartQueryXML	aimsQuery.js	Query/Find/Search
writeStoredQueryForm	aimsQuery.js	Query/Find/Search
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
zoomToReturnedRecords	aimsSelect.js	Graphic Selection

JavaScript functions by category

Category	File	Function Name
Basic Map	aimsCommon.js	checkCoords
Basic Map	aimsCommon.js	checkCurrentExtent
Basic Map	aimsCommon.js	checkForForbiddenTags
Basic Map	aimsCommon.js	clearError
Basic Map	aimsCommon.js	clearLeadingSpace
Basic Map	aimsCommon.js	convertDecimal
Basic Map	aimsCommon.js	convertHexToDec
Basic Map	aimsCommon.js	fixSingleQuotes
Basic Map	aimsCommon.js	formatDate
Basic Map	aimsCommon.js	getAllFieldValues
Basic Map	aimsCommon.js	getFieldNames
Basic Map	aimsCommon.js	getFieldValues
Basic Map	aimsCommon.js	getIDValue
Basic Map	aimsCommon.js	getInsideString
Basic Map	aimsCommon.js	getMapHeight
Basic Map	aimsCommon.js	getMapWidth
Basic Map	aimsCommon.js	getStartExtent
Basic Map	aimsCommon.js	justGetFeatureCount
Basic Map	aimsCommon.js	justGetFieldValue
Basic Map	aimsCommon.js	justGetValue
Basic Map	aimsCommon.js	makeXMLsafe
Basic Map	aimsCommon.js	numberorder
Basic Map	aimsCommon.js	parseEntity
Basic Map	aimsCommon.js	parseRecordString
Basic Map	aimsCommon.js	processStartExtent
Basic Map	aimsCommon.js	reloadApp
Basic Map	aimsCommon.js	replacePlus
Basic Map	aimsCommon.js	resetError
Basic Map	aimsCommon.js	startMap
Basic Map	aimsCommon.js	startUp
Basic Map	aimsCommon.js	swapQuotes
Basic Map	aimsCommon.js	swapStuff
Basic Map	aimsCommon.js	untag
Basic Map	aimsCommon.js	writeModeFrame
Basic Map	aimsCommon.js	writeModeLayers
Basic Map	aimsMap.js	afterMapRefresh
Basic Map	aimsMap.js	beforeMapRefresh
Basic Map	aimsMap.js	calcDistance
Basic Map	aimsMap.js	checkFullExtent

JavaScript functions by category (continued)

Category	File	Function Name
Basic Map	aimsMap.js	convertUnits
Basic Map	aimsMap.js	fullExtent
Basic Map	aimsMap.js	getCommandLineParams
Basic Map	aimsMap.js	getPath
Basic Map	aimsMap.js	getScaleBarDistance
Basic Map	aimsMap.js	hasLayers
Basic Map	aimsMap.js	hideRetrieveData
Basic Map	aimsMap.js	hideRetrieveMap
Basic Map	aimsMap.js	panButton
Basic Map	aimsMap.js	saveLastExtent
Basic Map	aimsMap.js	setExtent
Basic Map	aimsMap.js	setFullExtent
Basic Map	aimsMap.js	showRetrieveData
Basic Map	aimsMap.js	showRetrieveMap
Basic Map	aimsMap.js	startExtent
Basic Map	aimsMap.js	zoomBack
Basic Map	aimsMap.js	zoomButton
Basic Map	aimsMap.js	zoomScale
Basic Map	aimsMap.js	zoomToEnvelope
Basic Map	aimsMap.js	zoomToPoint
Basic Map	aimsXML.js	getEnvelopeXYs
Basic Map	aimsXML.js	getHost
Basic Map	aimsXML.js	getLegendURL
Basic Map	aimsXML.js	getOVXYs
Basic Map	aimsXML.js	getService
Basic Map	aimsXML.js	getURL
Basic Map	aimsXML.js	getXMLErrorMessage
Basic Map	aimsXML.js	getXYs
Basic Map	aimsXML.js	htmlSendToServer
Basic Map	aimsXML.js	isNotSameHostInURL
Basic Map	aimsXML.js	jspSendToServer
Basic Map	aimsXML.js	justGetMapextent
Basic Map	aimsXML.js	processXML
Basic Map	aimsXML.js	sendCustomToServer
Basic Map	aimsXML.js	sendMapXML
Basic Map	aimsXML.js	sendToServer
Basic Map	aimsXML.js	writeBlankMapXML
Basic Map	aimsXML.js	writeOVXML
Basic Map	aimsXML.js	writeXML

JavaScript functions by category (continued)

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

JavaScript functions by category (continued)

Category	File	Function Name
Geocode	aimsGeocode.js	getGeocodeLayers
Geocode	aimsGeocode.js	getGeocodeParams
Geocode	aimsGeocode.js	parseGeocodeLayers
Geocode	aimsGeocode.js	parseGeocodeParams
Geocode	aimsGeocode.js	setupGeocode
Geocode	aimsGeocode.js	writeGeocodeXML
Graphic Selection	aimsSelect.js	addSelectToMap
Graphic Selection	aimsSelect.js	calcSelectEnvelope
Graphic Selection	aimsSelect.js	clearSelection
Graphic Selection	aimsSelect.js	getMoreData
Graphic Selection	aimsSelect.js	select
Graphic Selection	aimsSelect.js	sendShapeSelect
Graphic Selection	aimsSelect.js	showHighlight
Graphic Selection	aimsSelect.js	startSelectBox
Graphic Selection	aimsSelect.js	stopSelectBox
Graphic Selection	aimsSelect.js	writeEnvelopeXML
Graphic Selection	aimsSelect.js	writeGetFeatures2
Graphic Selection	aimsSelect.js	writeGetFeatures3
Graphic Selection	aimsSelect.js	writeShapeSelect
Graphic Selection	aimsSelect.js	zoomToReturnedRecords
Identify/HyperLink	aimsIdentify.js	checkHyperLinkLayer
Identify/HyperLink	aimsIdentify.js	checkIfActiveLayerAvailable
Identify/HyperLink	aimsIdentify.js	checkSelected
Identify/HyperLink	aimsIdentify.js	displayAttributeData
Identify/HyperLink	aimsIdentify.js	displayAttributeDataforDrill
Identify/HyperLink	aimsIdentify.js	dIdentifyAll
Identify/HyperLink	aimsIdentify.js	hyperLink
Identify/HyperLink	aimsIdentify.js	hyperLinkAny
Identify/HyperLink	aimsIdentify.js	identify
Identify/HyperLink	aimsIdentify.js	identifyAll
Identify/HyperLink	aimsIdentify.js	parseHyperLink
Identify/HyperLink	aimsIdentify.js	parseHyperLinkAny
Identify/HyperLink	aimsIdentify.js	writeGetFeatures
Identify/HyperLink	aimsIdentify.js	writeGetFeaturesDrill
Identify/HyperLink	aimsIdentify.js	writIdentifyXML
Interactive Map	aimsClick.js	chkMouseUp
Interactive Map	aimsClick.js	clickAddPoint
Interactive Map	aimsClick.js	clickFunction
Interactive Map	aimsClick.js	deleteClick

JavaScript functions by category (continued)

Category	File	Function Name
Interactive Map	aimsClick.js	mapTool
Interactive Map	aimsClick.js	resetClick
Interactive Map	aimsClick.js	updateMeasureBox
Interactive Map	aimsNavigation.js	getImageXY
Interactive Map	aimsNavigation.js	getMapXY
Interactive Map	aimsNavigation.js	getMouse
Interactive Map	aimsNavigation.js	getOVImageXY
Interactive Map	aimsNavigation.js	ovMap2Click
Interactive Map	aimsNavigation.js	ovMapClick
Interactive Map	aimsNavigation.js	pan
Interactive Map	aimsNavigation.js	panMouse
Interactive Map	aimsNavigation.js	setClip
Interactive Map	aimsNavigation.js	setZoomColor
Interactive Map	aimsNavigation.js	startPan
Interactive Map	aimsNavigation.js	startZoomBox
Interactive Map	aimsNavigation.js	startZoomOutBox
Interactive Map	aimsNavigation.js	stopPan
Interactive Map	aimsNavigation.js	stopZoomBox
Interactive Map	aimsNavigation.js	stopZoomOutBox
Interactive Map	aimsNavigation.js	zoomin
Interactive Map	aimsNavigation.js	zoomout
Legend	aimsLegend.js	addLegendToMap
Legend	aimsLegend.js	getLegend
Legend	aimsLegend.js	showLegend
Print	aimsPrint.js	getPrintLegend
Print	aimsPrint.js	getPrintMap
Print	aimsPrint.js	getPrintOV
Print	aimsPrint.js	printIt
Print	aimsPrint.js	writePrintPage
Query/Find/Search	aimsQuery.js	checkStoredQueries
Query/Find/Search	aimsQuery.js	compare
Query/Find/Search	aimsQuery.js	findForm
Query/Find/Search	aimsQuery.js	getFind
Query/Find/Search	aimsQuery.js	getStoredQueries
Query/Find/Search	aimsQuery.js	parseFieldSamples
Query/Find/Search	aimsQuery.js	parseFieldSamplesUnique
Query/Find/Search	aimsQuery.js	parseStartQuery
Query/Find/Search	aimsQuery.js	parseStoredQueries
Query/Find/Search	aimsQuery.js	queryForm

JavaScript functions by category (continued)

Category	File	Function Name
Query/Find/Search	aimsQuery.js	sendQueryString
Query/Find/Search	aimsQuery.js	sendStoredQuery
Query/Find/Search	aimsQuery.js	setStartQuery
Query/Find/Search	aimsQuery.js	storedQueryForm
Query/Find/Search	aimsQuery.js	tempGetSamples
Query/Find/Search	aimsQuery.js	writeFieldSample
Query/Find/Search	aimsQuery.js	writeFindRequest
Query/Find/Search	aimsQuery.js	writeQueryForm
Query/Find/Search	aimsQuery.js	writeQueryXML
Query/Find/Search	aimsQuery.js	writeStartQueryXML
Query/Find/Search	aimsQuery.js	writeStoredQueryForm

addBufferToMap

Description:

Adds buffering instructions to ArcXML map image request.

Uses: bufferDistance, selectionMode, drawTargetLayer, bufferSmoothEdges, ScaleBarUnits, setQueryString, limitRight, limitLeft, limitBottom, 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, and highlightColor and clickPointX, clickY, and LayerName arrays.

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

Category:

Graphic Selection

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 hidelayer in aimsDHTML.js.

Called by: form created by writeBufferForm.

Category:

Buffer

File:

aimsBuffer.js

Syntax:

bufferIt()

Arguments:

None

Returned Value:

None

See Also:

writeBufferForm

hideLayer

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

calcSelectEnvelope

Description:

Calculates the minimum and maximum x and y coordinates and places the values in the selMaxEnvelope array.

Uses: selectCount as well as selMaxEnvelope, selectLeft, selectBottom, selectRight, and selectTop arrays.

Called by: zoomToReturnedRecords in aimsSelect.js.

Category:

Graphic Selection

File:

aimsSelect.js

Syntax:

calcSelectEnvelope()

Arguments:

None

Returned Value:

None

See Also:

zoomToReturnedRecords

checkCoords

Description:

Modifies NorthArrowCoords and CopyrightCoords with the value of coordsDelimiter.

Uses: NorthArrowCoords, CopyrightCoords, and coordsDelimiter.

Called by: checkParams in ArcIMSParam.js.

Category:

Basic Map

File:

aimsCommon.js

Syntax:

checkCoords()

Arguments:

None

Returned Value:

None

checkCurrentExtent

Description:

Displays current extent coordinates in an alert box.

Uses: eRight, eLeft, eTop, eBottom, xDistance, and fullWidth.

Used for: debugging.

Category:

Basic Map

File:

aimsCommon.js

Syntax:

checkCurrentExtent()

Arguments:

None

Returned Value:

None

checkForForbiddenTags

Description:

Checks for any ArcXML tags assigned to the service for the current user.

Uses: forbiddenTags, aimsSelectPresent, aimsQueryPresent, aimsBufferPresent, aimsIdentifyPresent, canQuery, useIdentify, useSelect, useQuery, useFind, useBuffer, useStoredQuery, useHyperLink, useHyperLinkAny, useIdentifyAll, useBufferShape, aimsGeocodePresent, useGeocode, useReverseGeocode, and useExtract.

Called by: processStartExtent in aimsCommon.js.

Category:

Basic Map

File:

aimsCommon.js

Syntax:

checkForForbiddenTags(theReply)

Arguments:

theReply Returned ArcXML response.

Returned Value:

None

See Also:

processStartExtent

checkFullExtent

Description:

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

Uses: eRight, eLeft, eTop, eBottom, 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:

Identify/HyperLink

File:

aimsIdentify.js

Syntax:

```
checkHyperLinkLayer(layerIndex)
```

Arguments:

layerIndex	Number representing the index of the layer. The top layer is zero.
------------	--

Returned Value:

Boolean	Returns true/false.
---------	---------------------

See Also:

clickFunction

checkIfActiveLayerAvailable

Description:

Checks to see if the Active Layer is available for action. Checks visibility and if the extent is within the minimum and maximum scale for the layer.

Uses: mapScaleFactor and ActiveLayerIndex as well as LayerMinScale, LayerMaxScale, LayerVisible, LayerName, and msgList arrays.

Called by: clickFunction in aimsClick.js; by identify and hyperLink in aimsIdentify; by queryForm, findForm, getStoredQueries, and storedQueryForm in aimsQuery; and by startSelectBox in aimsSelect.js.

Category:

Identify/HyperLink

File:

aimsIdentify.js

Syntax:

checkIfActiveLayerAvailable()

Arguments:

None

Returned Value:

Boolean If true, the Active Layer is available for action.

See Also:

clickFunction	findForm
identify	getStoredQueries
hyperLink	storedQueryForm
queryForm	startSelectBox

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:

<code>getPath</code>	<code>clickFunction</code>
<code>startMap</code>	<code>getCommonLineParams</code>

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:

Identify/HyperLink

File:

`aimsIdentify.js`

Syntax:

`checkSelected()`

Arguments:

None

Returned Value:

Boolean Returns true/false.

See Also:

`clickFunction`

checkStoredQueries

Description:

Checks if there are any StoredQueries in the service.

Sets: useStoredQuery to false if there are none.

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

Category:

Query/Find/Search

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 `aimsIdentify.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:

Graphic Selection

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, and legendVisible.

Calls: getMapXY, sendToServer, 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

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:

zoomToEnvelope	queryForm	printIt	extractIt
sendMapXML	fullExtent	setupGeocode	getLegend
writeModeFrame	zoomBack	getStoredQueries	hasLayer
writeModeLayers	showLayer	findForm	
deleteClick	updateMeasureBox	checkSelected	
hideLayer	resetClick	writeBufferForm	

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, putExtentOnOVMap, 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	putExtentOnOVMap
boxIt	panMouse
processXML	

compare

Description:

Function to compare numeric values for function parseFieldSamples numeric sort.

Called by: parseFieldSamplesUnique function.

Category:

Query

File:

aimsQuery.js

Syntax:

compare(a,b)

Arguments:

a First value.

b Second value.

Returned Value:

Numeric Lower value

See Also:

parseFieldSamplesUnique

convertDecimal

Description:

Formats decimal numbers using a comma to a point so they can be used by standard SQL queries. Certain languages use a comma instead of a point for decimals.

Called by: Query form created by writeQueryForm() in aimsQuery.js.

Category:

Basic Map

File:

aimsCommon.js

Syntax:

convertDecimal(theNumString)

Arguments:

theNumString String containing decimal numeric using a comma instead of a point.

Returned Value:

String Returns string containing decimal numeric using a point instead of a comma.

See Also:

writeQueryForm

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 (for example, “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.

Calls: sendToServer and writeXML.

Category:

Interactive Map

File:

aimsClick.js

Syntax:

deleteClick()

Arguments:

None

Returned Value:

None

See Also:

writeXML

sendToServer

displayAttributeDataforDrill

Description:

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

Uses: ActiveLayerIndex, useTextFrame, textFrameBackColor, tableBackColor, extWin, focusOnData, as well as charSet and titleList and replyArray arrays.

Calls: setLayerFields, writeOutDataPage, and hideRetrieveData.

Called by: doIdentifyAll in aimsIdentify.js.

Category:

Identify/HyperLink

File:

aimsIdentify.js

Syntax:

displayAttributeDataforDrill(theReplyArray)

Arguments:

theReplyArray	Array of strings containing returned ArcXML responses from query.
---------------	---

Returned Value:

None

See Also:

processXML	hideRetrieveData	setLayerFields
writeOutDataPage		

doIdentifyAll

Description:

Function used to request data from the list of visible available layers.

Uses: fID, idSouth, idNorth, idWest, idEast, and mapScaleFactor as well as LayerID, LayerMinScale, LayerMaxScale, LayerVisible, LayerIsFeature, and replyArray arrays.

Calls: writeGetFeaturesDrill, doIdentifyAll, and displayAttributeDataforDrill in aimsIdentify.js and sendToServer in aimsXML.js.

Called by: doIdentifyAll in aimsIdentify.js and processXML in aimsXML.js.

Category:

Identify/HyperLink

File:

aimsIdentify.js

Syntax:

doIdentifyAll(theReply)

Arguments:

theReply String containing returned ArcXML response.

Returned Value:

None

See Also:

writeGetFeaturesDrill	sendToServer
displayAttributeDataforDrill	processXML

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/Find/Search

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

formatDate

Description:

Converts a string containing a date into a format used in a query request.

Called by: Query form created by writeQueryForm() in aimsQuery.js.

Category:

Basic Map

File:

aimsCommon.js

Syntax:

formatDate(theDateString)

Arguments:

theDateString String containing a date.

Returned Value:

String Returns string containing date expression to be used in a query request in the format of a time stamp: "(ts 'yyyy-mm-dd hh:mm:ss')".

See Also:

writeQueryForm

fullExtent

Description:

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

Uses: aimsDHTMLPresent, hspc, vspc, eLeft, eRight, eTop, eBottom, fullLeft, fullRight, fullTop, 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.

Uses : bufferTargetLayerIndex, useTextFrame, textFrameBackColor, tableBackColor, extWin, focusOnData, as well as charSet and titleList array.

Calls: setLayerFields, writeOutDataPage, and hideRetrieveData.

Called by: processXML in aimsXML.js.

Category:

Buffer

File:

aimsBuffer.js

Syntax:

getBufferAttributeData(theReply)

Arguments:

theReply String containing returned ArcXML buffer response from query.

Returned Value:

None

See Also:

processXML

hideRetrieveData

setLayerFields

writeOutDataPage

getCommandLineParams

Description:

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

Uses: imsURL, imsOVURL, startLeft, startRight, startTop, 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, 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 `aimsIdentify.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 as well as 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` as well as `LayerIDField`, `LayerShapeField`, `LayerFields`, `LayerFieldType`, `showBuffer`, and `setQueryString` arrays.

Calls: `showRetrieveData` in `aimsMap.js`, `makeXMLsafe` in `aimsCommon.js`, `sendToServer` in `aimsXML.js`, and `writeFindRequest` in `aimsQuery.js`.

Called by: Find Form.

Category:

Query/Find/Search

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

getHost

Description:

Parses out host name from URL.

Called by: jspSendToServer function.

Category:

Basic Map

File:

aimsXML.js

Syntax:

getHost(theURL)

Arguments:

theURL	String containing URL.
--------	------------------------

Returned Value:

String	Host name from URL.
--------	---------------------

See Also:

jspSendToServer

getIDValue

Description:

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

Uses: ActiveLayerIndex and LayerIDField array.

Called by: displayAttributeData in aimsIdentify.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 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 top 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 top 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 LayerFieldSizeList 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 top 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 top 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 service.

Uses: layerCount and xmlEndPos.

Sets: ActiveLayerIndex, ActiveLayer, ActiveLayerType, and canQuery as well as 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, eLeft, and eBottom.

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

getMoreBufferData

Description:

Requests more records of the current selection set of features from a buffer target layer.

Sets: queryStartRecord.

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

Calls: writeQueryBufferXML, writeGetBufferFeatures, writeShapeBufferXML, showRetrieveData, and sendToServer.

Called by: getBufferAttributeData in aimsBuffer.js.

Category:

Buffer

File:

aimsBuffer.js

Syntax:

getMoreBufferData(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:

writeQueryBufferXML	writeGetBufferFeatures
writeShapeBufferXML	showRetrieveData
sendToServer	

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 aimsIdentify.js.

Category:

Graphic Selection

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, hasOVMap, ovIsVisible, ovMapIsLayer, mouseX, mouseY, i2Width, i2Height, iWidth, iHeight, zooming, selectBox, panning, x2, y2, pixelX, pixelY, xDistance, yDistance, eLeft, eBottom, 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 ArcIMSParam.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, fullOVRight, 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 ArcIMSParam.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, eLeft, eRight, eTop, eBottom, 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

getService

Description:

Parses out service name from URL.

Called by: jspSendToServer function.

Category:

Basic Map

File:

aimsXML.js

Syntax:

getService(theURL)

Arguments:

theURL String containing URL.

Returned Value:

String Service name from URL.

See Also:

jspSendToServer

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 service. Request is sent to server.

Uses: imsQueryURL.

Calls: sendToServer in aimsXML.js.

Called by: clickFunction in aimsClick.js.

Category:

Query/Find/Search

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, it calls getLegendURL for URL.

Uses: dQuote.

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

Called by: processXML in aimsXML.js.

Sets: theImageType 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 in aimsBuffer.js; clickFunction in aimsClick.js; useCustomFunction in 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, processStartExtent in aimsCommon.js, displayAttributeData and parseHyperLink in aimsIdentify.js, and 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:

<code>hideLayer</code>	<code>hasLayer</code>
<code>writePrintPage</code>	<code>sendToServer</code>

htmlSendToServer

Description:

Forwards ArcXML requests to Servlet Connector using HTML form loaded into the PostFrame frame.

Uses: localeEncoding.

Calls: isNotSameHostInURL in aimsXML.js.

Called by: sendToServer function.

Category:

Basic Map

File:

aimsXML.js

Syntax:

```
htmlSendToServer(URLString,XMLRequest,theType)
```

Arguments:

URLString	String containing URL.
XMLRequest	String containing ArcXML request.
theType	Numeric representing XML mode.

Returned Value:

None

See Also:

isNotSameHostInURL

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 as well as 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:

Identify/HyperLink

File:

aimsIdentify.js

Syntax:

hyperLink(e)

Arguments:

e Event from browser.

Returned Value:

None

See Also:

writeGetFeatures	getMapXY
showRetrieveData	sendToServer
clickFunction	

hyperLinkAny

Description:

Sends an ArcXML request for data to execute a hyperlink on the top visible layer with a defined hyperlink.

Uses: fIndex, fID, highlightedOne, mouseX, mouseY, searchTolerance, xDistance, iWidth, pixelTolerance, mapX, mapY, swapSelectFields, mapScaleFactor, imsQueryURL, and selectFields as well as the hyperLinkLayers, LayerMinScale, LayerMaxScale, LayerVisible, LayerIsFeature, LayerName, LayerIDField, LayerType, msgList, and LayerShapeField arrays.

Calls: writeIdentifyXML in aimsIdentify.js, showRetrieveData in aimsMap.js, getMapXY in aimsNavigation.js, and sendToServer in aimsXML.js.

Called by: mapTool in aimsClick.js.

Category:

Identify/HyperLink

File:

aimsIdentify.js

Syntax:

hyperLinkAny(e)

Arguments:

e Event from browser.

Returned Value:

None

See Also:

writeIdentifyXML	sendToServer
showRetrieveData	mapTool
getMapXY	

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:

Identify/HyperLink

File:

aimsIdentify.js

Syntax:

identify(e)

Arguments:

e Event from browser.

Returned Value:

None

See Also:

writeGetFeatures	showRetrieveData
sendToServer	clickFunction

identifyAll

Description:

Sends a sequence of ArcXML requests for data from the features of the visible layers at click point.

Uses: fID, highlightedOne, mouseX, mouseY, searchTolerance, xDistance, iWidth, pixelTolerance, idSouth, idNorth, idWest, idEast, and mapScaleFactor and LayerID, LayerMinScale, LayerMaxScale, LayerVisible, and LayerIsFeature arrays.

Calls: getMapXY in aimsNavigation.js, writeGetFeaturesDrill in aimsIdentify.js, showRetrieveData in aimsMap.js, and sendToServer in aimsXML.js.

Called by: mapTool in aimsClick.js.

Category:

Identify/HyperLink

File:

aimsIdentify.js

Syntax:

identifyAll(e)

Arguments:

e Event from browser.

Returned Value:

None

See Also:

mapTool	showRetrieveData
getMapXY	sendToServer
writeGetFeaturesDrill	

isNotSameHostInURL

Description:

Checks to see if the host in the URL for an ArcXML request is the same as a given host. Used in check for redirection of ArcXML request.

Called by: sendToServer in aimsXML.js.

Category:

Basic Map

File:

aimsXML.js

Syntax:

isNotSameHostInURL(theURL,theHost)

Arguments:

theURL	String containing URL to be checked.
theHost	String containing host name to be used in check.

Returned Value:

Boolean	If true, host in theURL is not the same as theHost.
---------	---

See Also:

sendToServer

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

jspSendToServer

Description:

Forwards ArcXML requests to Java Connector using HTML form loaded into the PostFrame frame. Requires that Java Connector be installed and Servlet Engine classpath include arcims_jconnect.jar.

Uses localeEncoding.

Calls getHost and getService in aimsXML.js.

Called by: sendToServer function.

Category:

Basic Map

File:

aimsXML.js

Syntax:

jspSendToServer(URLString,XMLRequest,theType)

Arguments:

URLString	String containing URL.
XMLRequest	String containing ArcXML request.
theType	Numeric representing XMLmode.

Returned Value:

None

See Also:

isNotSameHostInURL

sendToServer

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 aimsIdentify.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: eLeft, eRight, eTop, eBottom, and debugOn.

Calls: writeXML and sendToServer in aimsXML.js.

Category:

Basic Map

File:

aimsXML.js

Syntax:

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

Arguments:

theURL	String containing URL to service.
extentLeft	Numeric representing minimum map x coordinate.
extentTop	Numeric representing maximum map y coordinate.
extentRight	Numeric representing maximum map x coordinate.
extentBottom	Numeric representing minimum map y coordinate.
getOVMap	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 XML-compatible 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 mouse 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: putExtentOnOVMap 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	putExtentOnOVMap	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.

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, eLeft, eRight, eTop, and eBottom.

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

ovMap2Click

Description:

Function that executes on click in overview map display.

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

Calls: getOVImageXY 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:

getOVImageXY

ovMapClick

pan

Description:

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

Uses: mouseX, mouseY, lastLeft, lastRight, lastTop, lastBottom, eLeft, eRight, eTop, and eBottom.

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: eLeft, eRight, eTop, eBottom, 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. 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/Find/Search

File:

aimsQuery.js

Syntax:

parseFieldSamples(theReply)

Arguments:

theReply String containing returned ArcXML response.

Returned Value:

None

See Also:

justGetFeatureCount clearLeadingSpace

processXML

parseFieldSamplesUnique

Description:

Parses out unique values returned in Query samples request.

Uses selectData array.

Calls: compare function.

Called by: parseFieldSamples in aimsQuery.js.

Category:

Query

File:

aimsQuery.js

Syntax:

parseFieldSamplesUnique(theReply)

Arguments:

theReply String containing ArcXML response.

Returned Value:

None

See Also:

compare

parseFieldSamples

parseGeocodeLayers

Description:

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

Uses: dQuote.

Called by: processXML in aimsXML.js.

Sets: GCLayerCount as well as 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 as well as 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 and 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 as well as LayerName array.

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

Called by: processXML in aimsXML.js.

Category:

Identify/HyperLink

File:

aimsIdentify.js

Syntax:

parseHyperLink(theReply)

Arguments:

theReply String containing returned ArcXML response.

Returned Value:

None

See Also:

justGetFeatureCount	untag
hideRetrieveData	processXML

parseHyperLinkAny

Description:

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

Uses: dQuote, hyperlink WindowWidth, hyperlinkWindowHeight, currentHyperLinkField, newSelectCount, ActiveLayerIndex, and debugOn as well as hyperLinkLayers, hyperLinkFields, hyperLinkPrefix, hyperLinkSuffix, LayerMinScale, LayerMaxScale, LayerVisible, LayerIsFeature, LayerName, LayerIDField, LayerType, msgList, titleList, and LayerShapeField arrays.

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

Called by: parseHyperLinkAny in aimsIdentify.js and processXML in aimsXML.js.

Category:

Identify/HyperLink

File:

aimsIdentify.js

Syntax:

parseHyperLinkAny(theReply)

Arguments:

theReply String containing returned ArcXML response.

Returned Value:

None

See Also:

justGetFeatureCount processXML

hideRetrieveData

parseRecordString

Description:

Parses out record data from a returned ArcXML query response.

Sets: xmlEndPos.

Called by: displayAttributeData in aimsIdentify.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

parseStartQuery

Description:

Sends a query request on startup.

Uses: queryZoom, startLeft, startRight, startTop, startBottom, ActiveLayerIndex, fullWidth, selectPointMargin, fullHeight, selectMargin, selectType, eLeft, eRight, eTop, and eBottom as well as LayerType array.

Calls: getXMLErrorMessage, getXYs, and sendMapXML in aimsXML.js.

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

Category:

Query/Find/Search

File:

aimsQuery.js

Syntax:

parseStartQuery(theReply)

Arguments:

theReply String containing returned ArcXML response.

Returned Value:

None

See Also:

writeStartQueryXML	processStartExtent
sendToServer	processXML

parseStoredQueries

Description:

Parses out StoredQuery parameters from a returned ArcXML query response.

Uses: ActiveLayer and dQuote as well as storedQueryName, storedQueryString, storedQueryVariable, storedQueryVarCount, and storedQueryFieldList arrays.

Calls: storedQueryForm in aimsQuery.js.

Called by: processXML in aimsXML.js.

Sets: storedQueryCount and storedQueryIndex.

Category:

Query/Find/Search

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 in getLayerListContent.

Called by: processXML in aimsXML.js.

Sets: eLeft, eRight, eTop, eBottom, startLeft, startRight, startTop, startBottom, xDistance, yDistance, iWidth, iHeight, mapScaleFactor, xHalf, yHalf, panX, panY, panFactor, MapUnits, mouseX, mouseY, pixelX, pixelY, mapX, mapY, lastLeft, lastRight, lastTop, lastBottom, fullLeft, fullRight, fullTop, 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:

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, hasOVMap, ovVisible, pastStart, getBufferedData, aimsGeocodePresent, useGeocode, useReverseGeocode, drawLegendOnly, imsOVURL, imsURL, hspc, vspc, iWidth, iHeight, aimsClickPresent, clickCount, aimsBufferPresent, getLimitExtent, imsGeocodeURL, GSActiveLayer, appDir, fullLeft, fullRight, fullTop, fullBottom, eLeft, eRight, eTop, eBottom, limitLeft, limitRight, limitTop, limitBottom, fullOVLeft, fullOVRight, fullOVTop, fullOVBottom, fullWidth, fullHeight, fullOVWidth, fullOVHeight, useExternalWindow, useTextFrame, imageLimitLeft, imageLimitRight, imageLimitTop, and imageLimitBottom as well as 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:

replacePlus	getXYs	parseHyperLink
afterMapRefresh	updateMeasureBox	parseGeocodeLayers
getXMLErrorMessage	moveLayer	showLegend
getURL	clipLayer	getOVXYs
hasLayer	showRetrieveMap	writeOVXML
getLegendURL	displayAttributeData	sendToServer
getPrintOV	getBufferAttributeData	parseGeocodeParams
getPrintLegend	parseFieldSamples	putExtentOnOVMap
processStartExtent	hideRetrieveMap	writeGetBufferedData
writePrintPage	writeQueryForm	useCustomFunction
sendMapXML	parseStoredQueries	parseGeocodeResults

putExtentOnOVMap

Description:

Updates extent box in overview map display.

Uses: fullOVWidth, fullOVHeight, i2Width, i2Height, fullOVLeft, fullOVTop, eLeft, eRight, eTop, eBottom, ovBorderWidth, is5up, 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 as well as 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 in aimsClick.js.

Category:

Query/Find/Search

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: legendVisible, imsURL, blankImage, and toolMode.

Calls: sendToServer, writeXML, and updateMeasureBox.

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

resetPanImage

Description:

Repositions map image to default position and cancels onLoad event of map image. Called after user has panned map display.

Uses: toolMode, hspc, vspc, iWidth, and iHeight.

Calls: moveLayer, clipLayer2, and hasLayer.

Called by: onLoad event of map image when this function is declared as the event handler in stopPan.

Category:

Basic

File:

aimsXML.js

Syntax:

resetPanImage()

Arguments:

None

Returned Value:

None

See Also:

moveLayer	clipLayer2
hasLayer	stopPan

saveLastExtent

Description:

Function that saves the last map extent.

Uses: eLeft, eRight, eTop, and eBottom.

Called by: displayAttributeData in aimsIdentify.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 from 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 aimsIdentify.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 processXML 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 QueryServer.

Uses: setQueryString, selectionMode, ActiveLayerIndex, LayerFieldCount, showBuffer, ingsQueryURL, and queryXMLMode as well as 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/Find/Search

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 with user-created shape.

Uses: queryStartRecord, selectionMode, showBuffer, clickCount, imsQueryURL, and selectXMLMode as well as clickPointX and clickPointY arrays.

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

Called by: shape form in select.htm.

Category:

Graphic Selection

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 QueryServer.

Uses: imsQueryURL and queryXMLMode as well as 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/Find/Search

File:

aimsQuery.js

Syntax:

sendStoredQuery(theIndex,theValue)

Arguments:

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

theValue String containing user input value or values 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 or servers. This function updates the form in the PostFrame page (jsForm.htm) with the current request and processing function name.

Uses: okToSend, XMLMode, and debugOn.

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

Called by: functions sending ArcXML requests to the server or servers.

Category:

Basic Map

File:

aimsXML.js

Syntax:

sendToServer(URLString,XMLRequest,theType)

Arguments:

URLString String containing service 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:

showRetrieveMap hideRetrieveMap

hideRetrieveData

setActiveLayer

Description:

Sets the active layer.

Uses: queryOpen, useExternalWindow, tableBackColor, and useTextFrame as well as 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 top 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: eLeft, eRight, eTop, and eBottom.

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, fullRight, fullTop, 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:

getLayerset ZoomColor

setLayerFields

Description:

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

Uses: swapSelectFields and useFieldAlias as well as 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, and 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 top layer is zero.

Returned Value:

None

See Also:

getBufferAttributeData	displayAttributeData
setActiveLayer	queryForm
findForm	

setStartQuery

Description:

Sends a query request on startup.

Uses: highlightedOne and imsQueryURL.

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

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

Category:

Query/Find/Search

File:

aimsQuery.js

Syntax:

setStartQuery()

Arguments:

None

Returned Value:

None

See Also:

writeStartQueryXML

processStartExtent

sendToServer

processXML

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`

setupLayerVisible

Description:

Sets values of the elements of the LayerVisible array to the values of the elements in the setLayerVisible array. The setLayerVisible array is set up by getCommandLineParams in aimsMap.js using URL parameters.

Uses: LayerVisible and setLayerVisible arrays.

Called by: processStartExtent in aimsCommon.js.

Category:

Extended Map

File:

aimsLayers.js

Syntax:

setupLayerVisible()

Arguments:

None

Returned Value:

None

See Also:

processStartExtent

getCommandLineParams

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 as well as LayerIDField, selectPoints, selectLeft, selectRight, selectTop, and selectBottom arrays.

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

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

Sets: highlightedOne, eLeft, eRight, eTop, and eBottom.

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 as well as 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 top 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 aimsIdentify.js; sendQueryString, getFind, and sendStoredQuery in aimsQuery.js; and 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 GIF image while awaiting the response to a map request to the server.

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

Called by: getLegend in aimsLegend.js; getPrintMap in aimsPrint.js; and 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, eLeft, eRight, eTop, and eBottom.

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 service 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 ArcIMSParam.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:

Graphic Selection

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 service 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, eLeft, eRight, eTop, eBottom, 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, eLeft, eRight, eTop, eBottom, zleft, zright, ztop, 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:

Graphic Selection

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, eLeft, eRight, eTop, eBottom, zleft, zright, ztop, 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, eLeft, eRight, eTop, eBottom, zleft, zright, ztop, 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:

stopZoomOutBox(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/Find/Search

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/Find/Search

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, limitRight, limitTop, limitBottom, iWidth, iHeight, aimsLayersPresent, layerCount, and imsURL as well as 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 getBufferedData as well as 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, limitRight, limitTop, limitBottom, bufferDistance, bufferSmoothEdges, and ScaleBarUnits as well as selFieldList and LayerID arrays.

Called by: writeGetBufferedData 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:

writeGetBufferedData

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, limitRight, limitTop, and limitBottom.

Called by: writeGetFeatures2 and writeGetFeatures3 in aimsSelect.js.

Category:

Graphic Selection

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 as well as selectPoints, selectLeft, selectRight, selectTop, and selectBottom arrays.

Called by: queryForm and tempGetSamples in aimsQuery.js.

Category:

Query/Find/Search

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, limitRight, limitTop, limitBottom, selectLayer, selectType, ActiveLayerType, selectCount, and highlightedOne as well as selectPoints, selectLeft, selectRight, selectTop, and selectBottom arrays.

Called by: getFind in aimsQuery.js.

Category:

Query/Find/Search

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 as well as 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

writeGetBufferedData

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:

writeGetBufferedData()

Arguments:

None

Returned Value:

String ArcXML request string to be sent to QueryServer.

See Also:

writeQueryBufferXML

writeEnvelopeBufferXML

writeShapeBufferXML

processXML

writeGetBufferFeatures

Description:

Generates an ArcXML request for envelope spatial selection.

Uses: swapSelectFields, selectFields, bufferTargetLayerIndex, selectLayer, bufferTargetLayer, selectType, bufferTargetLayerIndex, selectCount, highlightedOne, selectPoints.length, selectField.length, selectRight.length, selectTop.length, selectBottom, and selectPoints as well as selFieldList array.

Calls: writeEnvelopeBufferXML.

Called by: getMoreBufferData in aimsBuffer.js.

Category:

Buffer

File:

aimsXML.js

Syntax:

writeGetBufferFeatures()

Arguments:

None

Returned Value:

None

See Also:

moveLayer	clipLayer2
hasLayer	stopPan

writeGetFeatures

Description:

Used by Identify and HyperLink modes to request data.

Uses: swapSelectFields, selectFields, ActiveLayerIndex, ActiveLayer, ActiveLayerType, useLimitExtent, and highlightedOne as well as 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 as well as selFieldList, selectPoints, selectLeft, selectRight, selectTop, and selectBottom arrays.

Calls: writeEnvelopeXML in aimsSelect.js.

Called by: select and stopSelectBox in aimsSelect.js.

Category:

Graphic Selection

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 as well as selFieldList, selectPoints, selectLeft, selectRight, selectTop, and selectBottom arrays.

Calls: writeEnvelopeXML in aimsSelect.js.

Called by: getMoreData in aimsSelect.js.

Category:

Graphic Selection

File:

aimsSelect.js

Syntax:

writeGetFeatures3()

Arguments:

None

Returned Value:

String ArcXML request string to be sent to the QueryServer.

See Also:

rightX Numeric representing selection envelope's right x coordinate.

writeGetFeaturesDrill

Description:

Used by identifyAll nodes to request data.

Uses: swapSelectFields, selectFields, ActiveLayerIndex, ActiveLayer, ActiveLayerType, useLimitExtent, and highlightedOne as well as selectFieldList and LayerId arrays.

Calls: writeIdentifyXML in aimsIdentify.js.

Called by: identifyAll and doIdentifyAll in aimsIdentify.js.

Category:

Identify/HyperLink

File:

aimsIdentify.js

Syntax:

```
writeGetFeaturesDrill(west1,south1,east1,north1,theID)
```

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 doIdentifyAll
identifyAll

writeIdentifyXML

Description:

Writes an ArcXML request for the Identify mode.

Uses: limitLeft, limitRight, limitTop, 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

writeModelLayers

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:

```
writeModelLayers(currentMode)
```

Arguments:

currentMode String containing name of current viewer mode.

Returned Value:

None

See Also:

replaceLayerContent clickFunction

writeOutDataPage

Description:

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

Uses: selectCount, showSelectedData, useExternalWindow, useTextFrame, toolMode, useFieldAlias, queryStartRecord, XMLEndPos, textFrameBackColor, tableBackColor, zoomToSingleSelect, selectPointMargin, selectMargin, and maxFeaturesReturned as well as selectLeft, selectRight, selectTop, selectBottom, hyperlinkLayers, hyperlinkFields, LayerName, selectPoints, AliasFieldName, LayerFields, LayerFieldType, and aliasFieldAlias arrays.

Calls: getXMLErrorMessage, justGetFeatureCount, getEnvelopeXYs, parseRecordString, getFieldNames, getFieldValues, and getIdValue.

Called by: displayAttributeData and displayAttributeDataforDrill in aimsIdentify.js, and getBufferAttributeData in aimsBuffer.js.

Category:

Identify/HyperLink

File:

aimsIdentify.js

Syntax:

```
writeOutDataPage(theReply,layerIndex,docObject,docName,pageObject)
```

Arguments:

theReply	String containing returned ArcXML response from query.
layerIndex	Index of layer.
docObject	Document object.
docName	Name of Document object.
pageObject	Page object.

Returned Value:

String	String containing attribute data for display to be written out in HTML format.
--------	--

See Also:

JustGetFeatureCount	getIdValue	getEnvelopeXYs	getXMLErrorMessage
getFieldNames	getFieldValues	parseRecordString	

writeOVXML

Description:

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

Uses: xDistance, fullWidth, fullHeight, eLeft, eRight, eTop, eBottom, xHalf, yHalf, fullOVLeft, fullOVTop, fullOVRight, fullOVBottom, i2Width, i2Height, toggleOVVisible, imsURL, imsOVURL, layerCount, mapBackColor, drawOVEntBox, 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, hasOVMap, 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, limitRight, limitTop, and limitBottom as well as selFieldList, LayerType, and LayerID arrays.

Called by: writeGetBufferedData 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:

writeGetBufferedData

writeQueryForm

Description:

Displays a query form for attribute queries.

Uses: fieldIndex, showSampleValues, dQuote, useTextFrame, textFrameBackColor, and LayerFieldCount as well as LayerFields, LayerFieldType, and selectData arrays.

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

Category:

Query/Find/Search

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, limitRight, limitTop, limitBottom, selectLayer, selectType, ActiveLayerType, selectCount, and highlightedOne as well as selFieldList, selectPoints, selectLeft, selectRight, selectTop, and selectBottom arrays.

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

Category:

Query/Find/Search

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, limitLeft, limitRight, limitTop, limitBottom, clickCount, bufferDistance, bufferSmoothEdges, and ScaleBarUnits as well as selFieldList, clickPointX, clickPointY, and LayerID arrays.

Called by: writeGetBufferedData 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:

writeGetBufferedData

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, limitLeft, limitRight, limitTop, limitBottom, clickCount, bufferDistance, bufferSmoothEdges, ScaleBarUnits, selectLayer, selectLayerType, selectCount, and highlightedOne as well as selFieldList, clickPointX, clickPointY, selectPoint, selectLeft, selectRight, selectTop, and selectBottom arrays.

Called by: sendShapeSelect and getMoreData in aimsSelect.js.

Category:

Graphic Selection

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
-----------------	-------------

writeStartQueryXML

Description:

Creates a query request string for startup.

Uses: swapSelectFields, selectFields, idAndShapeOnly, maxFeaturesReturned, LayerID array, and queryString.

Called by: setStartQuery in aimsQuery.js.

Category:

Query/Find/Search

File:

aimsQuery.js

Syntax:

```
writeStartQueryXML(queryString,idAndShapeOnly)
```

Arguments:

queryString	String containing SQL expression to be used in query request.
idAndShapeOnly	Boolean. If true, ID and shapefiles are the only fields to be returned in response.

Returned Value:

String	ArcXML request to be sent to the server.
--------	--

See Also:

setStartQuery

writeStoredQueryForm

Description:

Displays a query form for stored queries.

Uses: storedQueryIndex, useTextFrame, textFrameFormColor, ActiveLayerIndex, and storedQueryCount as well as LayerName, storedQueryName, and storedQueryVarCount arrays.

Calls: sendStoredQuery in aimsQuery.js.

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

Category:

Query/Find/Search

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: eLeft, eRight, eTop, eBottom, iHeight, iWidth, aimsLayersPresent, toggleVisible, layerCount, aimsClassRenderPresent, mapBackColor, aimsLegendPresent, legendVisible, aimsBufferPresent, showBuffer, aimsSelectPresent, showGeocode, geocodeX, geocodeY, geocodePointColor, geocodePointSize, geocodeLabel, aimsClickPresent, clickCount, selectColor, clickType, clickMarkerColor, clickMarkerType, clickMarkerSize, aimsCustomPresent, drawCopyright, CopyrightCoords, CopyrightText, CopyrightStyle, CopyrightSize, CopyrightFont, CopyrightColor, CopyrightBackground, CopyrightBGColor, CopyrightGlow, CopyrightGlowColor, drawNorthArrow, NorthArrowType, NorthArrowSize, NorthArrowCoords, NorthArrowAngle, drawScaleBar, ScaleBarBackColor, ScaleBarFont, ScaleBarStyle, ScaleBarColor, MapUnits, ScaleBarFontColor, ScaleBarUnits, ScaleBarPrecision, ScaleBarSize, ScaleBarWidth, drawModeOnMap, modeBlurb, modeMapColor, and modeMapGlow as well as 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; 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()

Arguments:

None

Returned Value:

String ArcXML request string to be sent to the ImageServer.

See Also:

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

zoomBack

Description:

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

Uses: aimsDHTMLPresent, eLeft, eRight, eTop, eBottom, 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: eLeft, eRight, eTop, eBottom, 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, eLeft, eRight, eTop, eBottom, 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, eLeft, eRight, eTop, eBottom, 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, eLeft, eRight, eTop, and eBottom.

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: eLeft, eRight, eBottom, and eTop.

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:

minXin Numeric representing envelope left x coordinate.

minYin Numeric representing envelope bottom y coordinate.

maxXin Numeric representing envelope right x coordinate.

maxYin 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, eLeft, eRight, eTop, eBottom, 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

zoomToReturnedRecords

Description:

Zooms main map display to area around the extent of returned features whose records are currently displayed.

Uses: fullWidth, fullHeight, selectPointMargin, xDistance, yDistance, selectMargin, legendTemp, and legendVisible as well as selMaxEnvelope array.

Calls: calcSelectEnvelope in aimsSelect.js, saveLastExtent in aimsMap.js, and sendMapXML in aimsXML.js.

Called by: link in page displaying returned selected features created by displayAttributeData in aimsIdentify.js.

Category:

Graphic Selection

File:

aimsSelect.js

Syntax:

zoomToReturnedRecords()

Arguments:

None

Returned Value:

None

See Also:

calcSelectEnvelope saveLastExtent
sendMapXML

Global variables by name

You edit global variables to influence the overall behavior of the viewer. They are stored in .js files; viewer behavior is organized into categories such as basic map and legend. Some of these categories are dependent on other categories. See ‘Organization of the HTML Viewer JavaScript Library’ for more information on the dependencies. Viewer behavior is further categorized into .js files. Each .js file controls a certain type of viewer behavior within the category. Each global variable is stored in a specific .js file; the variable can be used only by the functions in that same .js file. Use the following table to determine which .js file a particular variable is in. The subsequent table lists all the variables in each file.

Variable Name	File	Category
ActiveLayer	aimsLayers.js	Extended Map
ActiveLayerIndex	ArcIMSParam.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
aimsIdentifyPresent	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
aimsPrintPresent	MapFrame.htm	Basic Map
aimsQueryPresent	MapFrame.htm	Basic Map
aimsRoutePresent	MapFrame.htm	Basic Map
aimsSelectPresent	MapFrame.htm	Basic Map
aimsXMLPresent	MapFrame.htm	Basic Map
AliasFieldAliases	aimsLayers.js	Extended Map
AliasFieldNames	aimsLayers.js	Extended Map
allowOptions	ArcIMSParam.js	Basic Map
appDir	aimsMap.js	Basic Map
autoAdjustForArcMapServer	ArcIMSParam.js	Basic Map
blankImage	aimsClick.js	Interactive
bottomBarColor	ArcIMSParam.js	Basic Map
bottomBarHeight	ArcIMSParam.js	Basic Map
bottomBarOutline	ArcIMSParam.js	Basic Map
bufferDistance	aimsBuffer.js	Buffer
bufferSmoothEdges	aimsBuffer.js	Buffer
bufferTargetLayer	aimsBuffer.js	Buffer

Global variables by name (continued)

Variable Name	File	Category
bufferTargetLayerIndex	aimsBuffer.js	Buffer
buttonList	aimsResource.js	Basic Map
canLoad	aimsMap.js	Basic Map
canQuery	aimsIdentify.js	Identify/HyperLink
canSelectInvisible	aimsIdentify.js	Identify/HyperLink
catURL	ArcIMSparam.js	Basic Map
charEncoding	aimsXML.js	Basic Map
charSet	aimsXML.js	Basic Map
chkGeocodeLayers	aimsCommon.js	Basic Map
chkRouteLayers	aimsCommon.js	Basic Map
chkUnits	aimsCommon.js	Basic Map
ClassRenderLayer	ArcIMSparam.js	Basic Map
ClassRenderString	ArcIMSparam.js	Basic Map
clickCount	aimsClick.js	Interactive
clickFunction	aimsClick.js	Interactive
clickMarkerColor	ArcIMSparam.js	Basic Map
clickMarkerSize	ArcIMSparam.js	Basic Map
clickMarkerType	ArcIMSparam.js	Basic Map
clickMeasure	aimsClick.js	Interactive
clickPointX	aimsClick.js	Interactive
clickPointY	aimsClick.js	Interactive
clickType	aimsClick.js	Interactive
connectorType	viewer.htm	Basic Map
connectorURL	aimsXML.js	Basic Map
coordsDelimiter	aimsCommon.js	Basic Map
CopyrightBackground	ArcIMSparam.js	Basic Map
CopyrightBGColor	ArcIMSparam.js	Basic Map
CopyrightColor	ArcIMSparam.js	Basic Map
CopyrightCoords	ArcIMSparam.js	Basic Map
CopyrightFont	ArcIMSparam.js	Basic Map
CopyrightGlow	ArcIMSparam.js	Basic Map
CopyrightGlowColor	ArcIMSparam.js	Basic Map
CopyrightShadow	ArcIMSparam.js	Basic Map
CopyrightShadowColor	ArcIMSparam.js	Basic Map
CopyrightSize	ArcIMSparam.js	Basic Map
CopyrightStyle	ArcIMSparam.js	Basic Map
CopyrightText	ArcIMSparam.js	Basic Map
currentHyperLinkField	aimsIdentify.js	Identify/HyperLink
currentHyperLinkLayer	aimsIdentify.js	Identify/HyperLink

Global variables by name (continued)

Variable Name	File	Category
currentHyperLinkPrefix	aimsIdentify.js	Identify/HyperLink
currentHyperLinkSuffix	aimsIdentify.js	Identify/HyperLink
currentMeasure	aimsClick.js	Interactive
cVersion	aimsXML.js	Basic Map
debugOn	aimsMap.js	Basic Map
decimalChar	aimsCommon.js	Interactive
defaultLegTitle	aimsLegend.js	Legend
displayLayerInfoButton	aimsLayers.js	Extended Map
doURLencode	ArcIMSParam.js	Basic Map
dQuote	aimsMap.js	Basic Map
drawBottomBar	ArcIMSParam.js	Basic Map
drawCopyright	ArcIMSParam.js	Basic Map
drawFloatingMode	ArcIMSParam.js	Basic Map
drawLegendOnly	aimsLegend.js	Legend
drawModeOnMap	ArcIMSParam.js	Basic Map
drawNorthArrow	ArcIMSParam.js	Basic Map
drawOVExtentBox	aimsXML.js	Basic Map
drawScaleBar	ArcIMSParam.js	Basic Map
drawScaleBar2	ArcIMSParam.js	Basic Map
drawSelectBoundary	aimsCommon.js	Basic Map
drawTargetLayer	aimsBuffer.js	Buffer
eBottom	aimsMap.js	Basic Map
eLeft	aimsMap.js	Basic Map
enforceFullExtent	aimsMap.js	Basic Map
eRight	aimsMap.js	Basic Map
esriBlurb	ArcIMSParam.js	Basic Map
eTop	aimsMap.js	Basic Map
FeatureLayerCount	aimsLayers.js	Extended Map
flID	aimsIdentify.js	Identify/HyperLink
fieldAliasList	ArcIMSParam.js	Basic Map
fieldIndex	aimsLayers.js	Extended Map
findIndex	aimsIdentify.js	Identify/HyperLink
findXMLMode	aimsXML.js	Basic Map
forceNewOVMap	aimsXML.js	Basic Map
formColor	aimsXML.js	Basic Map
fullBottom	aimsMap.js	Basic Map
fullHeight	aimsMap.js	Basic Map
fullLeft	aimsMap.js	Basic Map
fullOVBottom	aimsMap.js	Basic Map

Global variables by name (continued)

Variable Name	File	Category
fullOVHeight	aimsMap.js	Basic Map
fullOVLeft	aimsMap.js	Basic Map
fullOVRight	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
GClablel	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
geocodeAppMode	aimsGeocode.js	Geocode
geocodeLabel	aimsMap.js	Basic Map
geocodeLabelSize	ArcIMSparam.js	Basic Map
geocodePointColor	ArcIMSparam.js	Basic Map
geocodePointSize	ArcIMSparam.js	Basic Map
geocodeX	aimsMap.js	Basic Map
geocodeY	aimsMap.js	Basic Map
getBufferedData	aimsBuffer.js	Buffer
getLayers	aimsLayers.js	Extended Map
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
hideIDFieldData	ArcIMSparam.js	Identify/HyperLink
hideLayersFromList	ArcIMSparam.js	Identify/HyperLink

Global variables by name (continued)

Variable Name	File	Category
hideShapeFieldData	ArcIMSparam.js	Identify/HyperLink
highlightColor	ArcIMSparam.js	Basic Map
highlightedOne	aimsSelect.js	Graphic Selection
hostName	ArcIMSparam.js	Basic Map
hspc	ArcIMSparam.js	Basic Map
hyperLinkFields	ArcIMSparam.js	Basic Map
hyperLinkLayers	ArcIMSparam.js	Basic Map
hyperLinkPrefix	ArcIMSparam.js	Basic Map
hyperLinkSuffix	ArcIMSparam.js	Basic Map
hyperlinkWindowHeight	aimsIdentify.js	Identify/HyperLink
hyperlinkWindowWidth	aimsIdentify.js	Identify/HyperLink
hyperlinkXMLMode	aimsXML.js	Basic Map
i2Height	aimsMap.js	Basic Map
i2Width	aimsMap.js	Basic Map
idEast	aimsIdentify.js	Identify/HyperLink
identifyXMLMode	aimsXML.js	Basic Map
idNorth	aimsIdentify.js	Identify/HyperLink
idSouth	aimsIdentify.js	Identify/HyperLink
idWest	aimsIdentify.js	Identify/HyperLink
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
is5up	ArcIMSparam.js	Basic Map
isArcMapService	aimsCommon.js	Basic Map
isIE	ArcIMSparam.js	Basic Map
isIE4	ArcIMSparam.js	Basic Map
isNav	ArcIMSparam.js	Basic Map
isNav4	ArcIMSparam.js	Basic Map
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

Global variables by name (continued)

Variable Name	File	Category
lastToMeasure	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
leftButton	aimsClick.js	Interactive
legendImage	aimsCommon.js	Basic Map
legendTemp	aimsCommon.js	Basic Map
legendVisible	aimsMap.js	Basic Map
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

Global variables by name (continued)

Variable Name	File	Category
listAllLayersInIDAll	ArcIMSparam.js	Basic Map
localeEncoding	aimsXML.js	Basic Map
mapBackColor	ArcIMSparam.js	Basic Map
mapScaleFactor	aimsMap.js	Basic Map
mapTransparent	ArcIMSparam.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
modeList	aimsResource.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
msgList	aimsResource.js	Basic Map
newSelectCount	aimsIdentify.js	Identify/HyperLink
noListLayer	ArcIMSparam.js	Basic Map
NorthArrowAngle	ArcIMSparam.js	Basic Map
NorthArrowCoords	ArcIMSparam.js	Basic Map
NorthArrowSize	ArcIMSparam.js	Basic Map
NorthArrowType	ArcIMSparam.js	Basic Map
numberDataSamples	ArcIMSparam.js	Basic Map
numDecimals	ArcIMSparam.js	Basic Map
numStatDecimals	ArcIMSparam.js	Basic Map
okToSend	aimsXML.js	Basic Map
onlyUniqueSamples	ArcIMSparam.js	Basic Map
ovBorderWidth	ArcIMSparam.js	Basic Map
ovBoxColor	ArcIMSparam.js	Basic Map
ovBoxSize	ArcIMSparam.js	Basic Map
ovExtentBoxSize	ArcIMSparam.js	Basic Map

Global variables by name (continued)

Variable Name	File	Category
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
pairsDelimiter	aimsCommon.js	Basic Map
panFactor	ArcIMSParam.js	Basic Map
panning	aimsClick.js	Interactive
panX	aimsMap.js	Basic Map
panY	aimsMap.js	Basic Map
parseGeocodeLayers	aimsGeocode.js	Geocode
parseGeocodeResults	aimsGeocode.js	Geocode
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
processStartExtent	aimsCommon.js	Basic Map
queryMode	aimsSelect.js	Graphic Selection
queryCaseInsensitive	ArcIMSParam.js	Basic Map
queryOpen	aimsLayers.js	Extended Map
queryStartRecord	aimsIdentify.js	Identify/HyperLink
queryXMLMode	aimsXML.js	Basic Map
queryZoom	aimsMap.js	Basic Map
replyArray	aimsIdentify.js	Identify/HyperLink
requestMethod	aimsXML.js	Basic Map
rightButton	aimsClick.js	Interactive
ScaleBarBackColor	ArcIMSParam.js	Basic Map
ScaleBarBackground	ArcIMSParam.js	Basic Map
ScaleBarColor	ArcIMSParam.js	Basic Map
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

Global variables by name (continued)

Variable Name	File	Category
ScaleBarUnits	ArcIMSParam.js	Basic Map
ScaleBarWidth	ArcIMSParam.js	Basic Map
ScaleBar2BackColor	ArcIMSParam.js	Basic Map
ScaleBar2Background	ArcIMSParam.js	Basic Map
ScaleBar2Color	ArcIMSParam.js	Basic Map
ScaleBar2Font	ArcIMSParam.js	Basic Map
ScaleBar2FontColor	ArcIMSParam.js	Basic Map
ScaleBar2Precision	ArcIMSParam.js	Basic Map
ScaleBar2Round	ArcIMSParam.js	Basic Map
ScaleBar2Size	ArcIMSParam.js	Basic Map
ScaleBar2Style	ArcIMSParam.js	Basic Map
ScaleBar2Units	ArcIMSParam.js	Basic Map
ScaleBar2Width	ArcIMSParam.js	Basic Map
SdcGeocodeStyle	aimsGeocode.js	Geocode
searchTolerance	aimsIdentify.js	Identify/HyperLink
selectBlurb	aimsSelect.js	Graphic Selection
selectBottom	aimsSelect.js	Graphic Selection
selectBox	aimsClick.js	Interactive
selectColor	ArcIMSParam.js	Basic Map
selectCount	aimsSelect.js	Graphic Selection
selectData	aimsSelect.js	Graphic Selection
selectEnvelope	aimsSelect.js	Graphic Selection
selectFields	ArcIMSParam.js	Basic Map
selectionMode	aimsSelect.js	Graphic Selection
selectLayer	aimsSelect.js	Graphic Selection
selectLeft	aimsSelect.js	Graphic Selection
selectMargin	ArcIMSParam.js	Basic Map
selectPointMargin	ArcIMSParam.js	Basic Map
selectPoints	aimsIdentify.js	Identify/HyperLink
selectRight	aimsSelect.js	Graphic Selection
selectTop	aimsSelect.js	Graphic Selection
selectType	aimsSelect.js	Graphic Selection
selectXMLMode	aimsXML.js	Basic Map
selFieldList	ArcIMSParam.js	Basic Map
selMaxEnvelope	aimsSelect.js	Graphic Selection
serverURL	ArcIMSParam.js	Basic Map
setDebug	aimsMap.js	Basic Map
setLayerVisible	aimsLayers.js	Extended Map
setMapUnits	ArcIMSParam.js	Basic Map

Global variables by name (continued)

Variable Name	File	Category
setQueryString	aimsSelect.js	Graphic Selection
shapeBufferDistance	aimsSelect.js	Graphic Selection
shapeSelectBuffer	aimsClick.js	Interactive
showBuffer	aimsCommon.js	Basic Map
showGeocode	aimsMap.js	Basic Map
showSampleValues	aimsQuery.js	Query/Find/Search
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/Find/Search
storedQueryFieldList	aimsQuery.js	Query/Find/Search
storedQueryIndex	aimsQuery.js	Query/Find/Search
storedQueryName	aimsQuery.js	Query/Find/Search
storedQueryString	aimsQuery.js	QueryFind/Search
storedQueryVarCount	aimsQuery.js	Query/Find/Search
storedQueryVariable	aimsQuery.js	Query/Find/Search
sUnitList	aimsResource.js	Basic Map
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
theCursor	aimsMap.js	Basic Map
theImageType	aimsXML.js	Basic Map
titleList	aimsResource.js	Basic Map
toggleOVVisible	ArcIMSparam.js	Basic Map
toggleVisible	ArcIMSparam.js	Basic Map
toolMode	aimsMap.js	Basic Map
totalMeasure	aimsClick.js	Interactive
transColor	ArcIMSparam.js	Basic Map
transparentLevel	ArcIMSparam.js	Basic Map

Global variables by name (continued)

Variable Name	File	Category
unitList	aimsResource.js	Basic Map
useBuffer	ArcIMSParam.js	Basic Map
useBufferShape	ArcIMSParam.js	Basic Map
useClearSelect	ArcIMSParam.js	Basic Map
useExternalWindow	ArcIMSParam.js	Basic Map
useExtract	ArcIMSParam.js	Basic Map
useFieldAlias	ArcIMSParam.js	Basic Map
useFind	ArcIMSParam.js	Basic Map
useFullExtent	ArcIMSParam.js	Basic Map
useGeocode	ArcIMSParam.js	Basic Map
useHyperLink	ArcIMSParam.js	Basic Map
useHyperLinkAny	ArcIMSParam.js	Basic Map
useIdentify	ArcIMSParam.js	Basic Map
useIdentifyAll	ArcIMSParam.js	Basic Map
useLimitExtent	aimsMap.js	Basic Map
useMeasure	ArcIMSParam.js	Basic Map
useModeFrame	ArcIMSParam.js	Basic Map
usePan	ArcIMSParam.js	Basic Map
usePanEast	ArcIMSParam.js	Basic Map
usePanNorth	ArcIMSParam.js	Basic Map
usePanSouth	ArcIMSParam.js	Basic Map
usePanWest	ArcIMSParam.js	Basic Map
usePrint	ArcIMSParam.js	Basic Map
useQuery	ArcIMSParam.js	Basic Map
useReverseGeocode	ArcIMSParam.js	Basic Map
useRoute	ArcIMSParam.js	Basic Map
useSelect	ArcIMSParam.js	Basic Map
useSetUnits	ArcIMSParam.js	Basic Map
useStoredQuery	ArcIMSParam.js	Basic Map
useTextFrame	ArcIMSParam.js	Basic Map
useZoomActive	ArcIMSParam.js	Basic Map
useZoomIn	ArcIMSParam.js	Basic Map
useZoomLast	ArcIMSParam.js	Basic Map
useZoomOut	ArcIMSParam.js	Basic Map
vspc	ArcIMSParam.js	Basic Map
webParams	ArcIMSParam.js	Basic Map
writeXML	aimsXML.js	Basic Map
x1	aimsClick.js	Interactive
x2	aimsClick.js	Interactive

Global variables by name (continued)

Variable Name	File	Category
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	ArcIMSParam.js	Basic Map
zoomFactor	ArcIMSParam.js	Basic Map
zooming	aimsClick.js	Interactive
zoomToSingleSelect	ArcIMSParam.js	Basic Map
zright	aimsClick.js	Interactive
ztop	aimsClick.js	Interactive

Global variables by category

Category	File	Variable Name
Basic Map	aimsCommon.js	chkGeocodeLayers
Basic Map	aimsCommon.js	chkRouteLayers
Basic Map	aimsCommon.js	chkUnits
Basic Map	aimsCommon.js	coordsDelimiter
Basic Map	aimsCommon.js	drawSelectBoundary
Basic Map	aimsCommon.js	isArcMapService
Basic Map	aimsCommon.js	legendImage
Basic Map	aimsCommon.js	legendTemp
Basic Map	aimsCommon.js	modeBlurb
Basic Map	aimsCommon.js	ovlsVisible
Basic Map	aimsCommon.js	pairsDelimiter
Basic Map	aimsCommon.js	showBuffer
Basic Map	aimsMap.js	appDir
Basic Map	aimsMap.js	canLoad
Basic Map	aimsMap.js	debugOn
Basic Map	aimsMap.js	dQuote
Basic Map	aimsMap.js	eBottom
Basic Map	aimsMap.js	eLeft
Basic Map	aimsMap.js	enforceFullExtent
Basic Map	aimsMap.js	eRight
Basic Map	aimsMap.js	eTop
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	fullOVRight
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
Basic Map	aimsMap.js	geocodeY
Basic Map	aimsMap.js	getLimitExtent
Basic Map	aimsMap.js	getStartingExtent
Basic Map	aimsMap.js	i2Height

Global variables by category (continued)

Category	File	Variable Name
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	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	queryZoom
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	useLimitExtent
Basic Map	aimsMap.js	xDistance
Basic Map	aimsMap.js	yDistance
Basic Map	aimsResource.js	buttonList
Basic Map	aimsResource.js	modeList
Basic Map	aimsResource.js	msgList
Basic Map	aimsResource.js	sUnitList
Basic Map	aimsResource.js	titleList
Basic Map	aimsResource.js	unitList
Basic Map	aimsXML.js	charEncoding
Basic Map	aimsXML.js	charSet
Basic Map	aimsXML.js	connectorURL
Basic Map	aimsXML.js	cVersion
Basic Map	aimsXML.js	drawOVExtentBox

Global variables by category (continued)

Category	File	Variable Name
Basic Map	aimsXML.js	findXMLMode
Basic Map	aimsXML.js	forceNewOVMap
Basic Map	aimsXML.js	formColor
Basic Map	aimsXML.js	hyperlinkXMLMode
Basic Map	aimsXML.js	identifyXMLMode
Basic Map	aimsXML.js	localeEncoding
Basic Map	aimsXML.js	okToSend
Basic Map	aimsXML.js	pastStart
Basic Map	aimsXML.js	queryXMLMode
Basic Map	aimsXML.js	requestMethod
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
Basic Map	ArcIMSParam.js	ActiveLayerIndex
Basic Map	ArcIMSParam.js	allowOptions
Basic Map	ArcIMSParam.js	bottomBarColor
Basic Map	ArcIMSParam.js	bottomBarHeight
Basic Map	ArcIMSParam.js	bottomBarOutline
Basic Map	ArcIMSParam.js	catURL
Basic Map	ArcIMSParam.js	ClassRenderLayer
Basic Map	ArcIMSParam.js	ClassRenderString
Basic Map	ArcIMSParam.js	clickMarkerColor
Basic Map	ArcIMSParam.js	clickMarkerSize
Basic Map	ArcIMSParam.js	clickMarkerType
Basic Map	ArcIMSParam.js	CopyrightBackground
Basic Map	ArcIMSParam.js	CopyrightBGColor
Basic Map	ArcIMSParam.js	CopyrightColor
Basic Map	ArcIMSParam.js	CopyrightCoords
Basic Map	ArcIMSParam.js	CopyrightFont
Basic Map	ArcIMSParam.js	CopyrightGlow
Basic Map	ArcIMSParam.js	CopyrightGlowColor
Basic Map	ArcIMSParam.js	CopyrightShadow
Basic Map	ArcIMSParam.js	CopyrightShadowColor
Basic Map	ArcIMSParam.js	CopyrightSize
Basic Map	ArcIMSParam.js	CopyrightStyle
Basic Map	ArcIMSParam.js	CopyrightText

Global variables by category (continued)

Category	File	Variable Name
Basic Map	ArcIMSparam.js	doURLencode
Basic Map	ArcIMSparam.js	autoAdjustForArcMapServer
Basic Map	ArcIMSparam.js	drawBottomBar
Basic Map	ArcIMSparam.js	drawCopyright
Basic Map	ArcIMSparam.js	drawFloatingMode
Basic Map	ArcIMSparam.js	drawModeOnMap
Basic Map	ArcIMSparam.js	drawNorthArrow
Basic Map	ArcIMSparam.js	drawScaleBar
Basic Map	ArcIMSparam.js	drawScaleBar2
Basic Map	ArcIMSparam.js	esriBlurb
Basic Map	ArcIMSparam.js	fieldAliasList
Basic Map	ArcIMSparam.js	geocodeLabelSize
Basic Map	ArcIMSparam.js	geocodePointColor
Basic Map	ArcIMSparam.js	geocodePointSize
Basic Map	ArcIMSparam.js	hasOVMap
Basic Map	ArcIMSparam.js	hasTOC
Basic Map	ArcIMSparam.js	hasToolBarOnLayer
Basic Map	ArcIMSparam.js	hideIDFieldData
Basic Map	ArcIMSparam.js	hideLayersFromList
Basic Map	ArcIMSparam.js	hideShapeFieldData
Basic Map	ArcIMSparam.js	highlightColor
Basic Map	ArcIMSparam.js	hostName
Basic Map	ArcIMSparam.js	hspc
Basic Map	ArcIMSparam.js	hyperLinkFields
Basic Map	ArcIMSparam.js	hyperLinkLayers
Basic Map	ArcIMSparam.js	hyperLinkPrefix
Basic Map	ArcIMSparam.js	hyperLinkSuffix
Basic Map	ArcIMSparam.js	imsGeocodeURL
Basic Map	ArcIMSparam.js	imsOVURL
Basic Map	ArcIMSparam.js	imsQueryURL
Basic Map	ArcIMSparam.js	imsURL
Basic Map	ArcIMSparam.js	is5up
Basic Map	ArcIMSparam.js	isIE
Basic Map	ArcIMSparam.js	isIE4
Basic Map	ArcIMSparam.js	isNav
Basic Map	ArcIMSparam.js	isNav4
Basic Map	ArcIMSparam.js	legFont
Basic Map	ArcIMSparam.js	legHeight
Basic Map	ArcIMSparam.js	legTitle

Global variables by category (continued)

Category	File	Variable Name
Basic Map	ArcIMSparam.js	legWidth
Basic Map	ArcIMSparam.js	limitBottom
Basic Map	ArcIMSparam.js	limitLeft
Basic Map	ArcIMSparam.js	limitRight
Basic Map	ArcIMSparam.js	limitTop
Basic Map	ArcIMSparam.js	listAllLayers
Basic Map	ArcIMSparam.js	listAllLayersInIDAll
Basic Map	ArcIMSparam.js	mapBackColor
Basic Map	ArcIMSparam.js	mapTransparent
Basic Map	ArcIMSparam.js	MapUnits
Basic Map	ArcIMSparam.js	maxFeaturesReturned
Basic Map	ArcIMSparam.js	maxGeocodeCandidates
Basic Map	ArcIMSparam.js	minGeocodeScore
Basic Map	ArcIMSparam.js	modeLayerColor
Basic Map	ArcIMSparam.js	modeLayerFont
Basic Map	ArcIMSparam.js	modeLayerOn
Basic Map	ArcIMSparam.js	modeLayerShadowColor
Basic Map	ArcIMSparam.js	modeLayerSize
Basic Map	ArcIMSparam.js	modeMapColor
Basic Map	ArcIMSparam.js	modeMapGlow
Basic Map	ArcIMSparam.js	modeRefreshMap
Basic Map	ArcIMSparam.js	noListLayer
Basic Map	ArcIMSparam.js	NorthArrowAngle
Basic Map	ArcIMSparam.js	NorthArrowCoords
Basic Map	ArcIMSparam.js	NorthArrowSize
Basic Map	ArcIMSparam.js	NorthArrowType
Basic Map	ArcIMSparam.js	numberDataSamples
Basic Map	ArcIMSparam.js	numDecimals
Basic Map	ArcIMSparam.js	numStatDecimals
Basic Map	ArcIMSparam.js	onlyUniqueSamples
Basic Map	ArcIMSparam.js	ovBorderWidth
Basic Map	ArcIMSparam.js	ovBoxColor
Basic Map	ArcIMSparam.js	ovBoxSize
Basic Map	ArcIMSparam.js	ovExtentBoxSize
Basic Map	ArcIMSparam.js	ovHspc
Basic Map	ArcIMSparam.js	ovImageVar
Basic Map	ArcIMSparam.js	ovMapIsLayer
Basic Map	ArcIMSparam.js	ovVspc
Basic Map	ArcIMSparam.js	panFactor

Global variables by category (continued)

Category	File	Variable Name
Basic Map	ArcIMSparam.js	pixelTolerance
Basic Map	ArcIMSparam.js	queryCaseInsensitive
Basic Map	ArcIMSparam.js	ScaleBar2BackColor
Basic Map	ArcIMSparam.js	ScaleBar2Background
Basic Map	ArcIMSparam.js	ScaleBar2Color
Basic Map	ArcIMSparam.js	ScaleBar2Font
Basic Map	ArcIMSparam.js	ScaleBar2FontColor
Basic Map	ArcIMSparam.js	ScaleBar2Precision
Basic Map	ArcIMSparam.js	ScaleBar2Round
Basic Map	ArcIMSparam.js	ScaleBar2Size
Basic Map	ArcIMSparam.js	ScaleBar2Style
Basic Map	ArcIMSparam.js	ScaleBar2Units
Basic Map	ArcIMSparam.js	ScaleBar2Width
Basic Map	ArcIMSparam.js	ScaleBarBackColor
Basic Map	ArcIMSparam.js	ScaleBarBackground
Basic Map	ArcIMSparam.js	ScaleBarColor
Basic Map	ArcIMSparam.js	ScaleBarFont
Basic Map	ArcIMSparam.js	ScaleBarFontColor
Basic Map	ArcIMSparam.js	ScaleBarPrecision
Basic Map	ArcIMSparam.js	ScaleBarRound
Basic Map	ArcIMSparam.js	ScaleBarSize
Basic Map	ArcIMSparam.js	ScaleBarStyle
Basic Map	ArcIMSparam.js	ScaleBarUnits
Basic Map	ArcIMSparam.js	ScaleBarWidth
Basic Map	ArcIMSparam.js	selectColor
Basic Map	ArcIMSparam.js	selectFields
Basic Map	ArcIMSparam.js	selectMargin
Basic Map	ArcIMSparam.js	selectPointMargin
Basic Map	ArcIMSparam.js	selfFieldList
Basic Map	ArcIMSparam.js	serverURL
Basic Map	ArcIMSparam.js	setMapUnits
Basic Map	ArcIMSparam.js	showScalePercent
Basic Map	ArcIMSparam.js	showSelectedData
Basic Map	ArcIMSparam.js	showSelectedFeatures
Basic Map	ArcIMSparam.js	showTOC
Basic Map	ArcIMSparam.js	showXYs
Basic Map	ArcIMSparam.js	startBottom
Basic Map	ArcIMSparam.js	startLeft
Basic Map	ArcIMSparam.js	startRight

Global variables by category (continued)

Category	File	Variable Name
Basic Map	ArcIMSparam.js	startTop
Basic Map	ArcIMSparam.js	swapSelectFields
Basic Map	ArcIMSparam.js	textBackColor
Basic Map	ArcIMSparam.js	textFrameBackColor
Basic Map	ArcIMSparam.js	textFrameFormColor
Basic Map	ArcIMSparam.js	textFrameLinkColor
Basic Map	ArcIMSparam.js	textFrameTextColor
Basic Map	ArcIMSparam.js	toggleOVVisible
Basic Map	ArcIMSparam.js	toggleVisible
Basic Map	ArcIMSparam.js	transColor
Basic Map	ArcIMSparam.js	transparentLevel
Basic Map	ArcIMSparam.js	useBuffer
Basic Map	ArcIMSparam.js	useBufferShape
Basic Map	ArcIMSparam.js	useClearSelect
Basic Map	ArcIMSparam.js	useExternalWindow
Basic Map	ArcIMSparam.js	useExtract
Basic Map	ArcIMSparam.js	useFieldAlias
Basic Map	ArcIMSparam.js	useFind
Basic Map	ArcIMSparam.js	useFullExtent
Basic Map	ArcIMSparam.js	useGeocode
Basic Map	ArcIMSparam.js	useHyperLink
Basic Map	ArcIMSparam.js	useHyperLinkAny
Basic Map	ArcIMSparam.js	useIdentify
Basic Map	ArcIMSparam.js	useIdentifyAll
Basic Map	ArcIMSparam.js	useMeasure
Basic Map	ArcIMSparam.js	useModeFrame
Basic Map	ArcIMSparam.js	usePan
Basic Map	ArcIMSparam.js	usePanEast
Basic Map	ArcIMSparam.js	usePanNorth
Basic Map	ArcIMSparam.js	usePanSouth
Basic Map	ArcIMSparam.js	usePanWest
Basic Map	ArcIMSparam.js	usePrint
Basic Map	ArcIMSparam.js	useQuery
Basic Map	ArcIMSparam.js	useReverseGeocode
Basic Map	ArcIMSparam.js	useRoute
Basic Map	ArcIMSparam.js	useSelect
Basic Map	ArcIMSparam.js	useSetUnits
Basic Map	ArcIMSparam.js	useStoredQuery
Basic Map	ArcIMSparam.js	useTextFrame

Global variables by category (continued)

Category	File	Variable Name
Basic Map	ArcIMSParam.js	useZoomActive
Basic Map	ArcIMSParam.js	useZoomIn
Basic Map	ArcIMSParam.js	useZoomLast
Basic Map	ArcIMSParam.js	useZoomOut
Basic Map	ArcIMSParam.js	vspc
Basic Map	ArcIMSParam.js	webParams
Basic Map	ArcIMSParam.js	zoomBoxColor
Basic Map	ArcIMSParam.js	zoomFactor
Basic Map	ArcIMSParam.js	zoomToSingleSelect
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	aimsIdentifyPresent
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	viewer.htm	connectorType
Buffer	aimsBuffer.js	bufferDistance
Buffer	aimsBuffer.js	bufferSmoothEdges
Buffer	aimsBuffer.js	bufferTargetLayer
Buffer	aimsBuffer.js	bufferTargetLayerIndex
Buffer	aimsBuffer.js	drawTargetLayer
Buffer	aimsBuffer.js	getBufferedData
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

Global variables by category (continued)

Category	File	Variable Name
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	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	layersBottom
Extended Map	aimsLayers.js	LayerShapeField
Extended Map	aimsLayers.js	layersLeft
Extended Map	aimsLayers.js	layersRight
Extended Map	aimsLayers.js	layersTop
Extended Map	aimsLayers.js	LayerType
Extended Map	aimsLayers.js	LayerVisible
Extended Map	aimsLayers.js	queryOpen
Extended Map	aimsLayers.js	setLayerVisible
Geocode	aimsGeocode.js	GCActiveLayer
Geocode	aimsGeocode.js	GCaddress
Geocode	aimsGeocode.js	GCdesc
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

Global variables by category (continued)

Category	File	Variable Name
Geocode	aimsGeocode.js	GCscore
Geocode	aimsGeocode.js	GCvalue
Geocode	aimsGeocode.js	geocodeAppMode
Geocode	aimsGeocode.js	SdcGeocodeStyle
Graphic Selection	aimsSelect.js	highlightedOne
Graphic Selection	aimsSelect.js	queryMode
Graphic Selection	aimsSelect.js	selectBlurb
Graphic Selection	aimsSelect.js	selectBottom
Graphic Selection	aimsSelect.js	selectCount
Graphic Selection	aimsSelect.js	selectData
Graphic Selection	aimsSelect.js	selectEnvelope
Graphic Selection	aimsSelect.js	selectionMode
Graphic Selection	aimsSelect.js	selectLayer
Graphic Selection	aimsSelect.js	selectLeft
Graphic Selection	aimsSelect.js	selectRight
Graphic Selection	aimsSelect.js	selectTop
Graphic Selection	aimsSelect.js	selectType
Graphic Selection	aimsSelect.js	selMaxEnvelope
Graphic Selection	aimsSelect.js	setQueryString
Graphic Selection	aimsSelect.js	shapeBufferDistance
Identify/HyperLink	aimsIdentify.js	canQuery
Identify/HyperLink	aimsIdentify.js	canSelectInvisible
Identify/HyperLink	aimsIdentify.js	currentHyperLinkField
Identify/HyperLink	aimsIdentify.js	currentHyperLinkLayer
Identify/HyperLink	aimsIdentify.js	currentHyperLinkPrefix
Identify/HyperLink	aimsIdentify.js	currentHyperLinkSuffix
Identify/HyperLink	aimsIdentify.js	fIndex
Identify/HyperLink	aimsIdentify.js	hyperlinkWindowHeight
Identify/HyperLink	aimsIdentify.js	hyperlinkWindowWidth
Identify/HyperLink	aimsIdentify.js	idEast
Identify/HyperLink	aimsIdentify.js	idNorth
Identify/HyperLink	aimsIdentify.js	idSouth
Identify/HyperLink	aimsIdentify.js	idWest
Identify/HyperLink	aimsIdentify.js	newSelectCount
Identify/HyperLink	aimsIdentify.js	queryStartRecord
Identify/HyperLink	aimsIdentify.js	replyArray
Identify/HyperLink	aimsIdentify.js	searchTolerance
Identify/HyperLink	aimsIdentify.js	selectPoints
Identify/HyperLink	ArcIMSparam.js	hideIDFieldData

Global variables by category (continued)

Category	File	Variable Name
Identify/HyperLink	ArcIMSparam.js	hideLayersFromList
Identify/HyperLink	ArcIMSparam.js	hideShapeFieldData
Identify/HyperLink	ArcIMSparam.js	hyperLinkPrefix
Identify/HyperLink	ArcIMSparam.js	hyperLinkSuffix
Identify/HyperLink	ArcIMSparam.js	listAllLayersInIDAll
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	shapeSelectBuffer
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
Interactive	aimsCommon.js	decimalChar
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

Global variables by category (continued)

Category	File	Variable Name
Query/Find/Search	aimsQuery.js	storedQueryCount
Query/Find/Search	aimsQuery.js	storedQueryFieldList
Query/Find/Search	aimsQuery.js	storedQueryIndex
Query/Find/Search	aimsQuery.js	storedQueryName
Query/Find/Search	aimsQuery.js	storedQueryString
Query/Find/Search	aimsQuery.js	storedQueryVariable
Query/Find/Search	aimsQuery.js	showSampleValues

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.
getBufferedData	(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) Filepath 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.
clickFunction	(String) The writeXML function in aimsXML.js checks the value in this function. A value of route supports the call that starts the routing mode. A value of address supports the call to start the reverse-geocoding mode.
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.
GCActiveLayer	(String/Dynamic) ID value of current active geocode layer. Updated by viewer with each geocode request.
lastToMeasure	(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 pointer. Updated by viewer on mouse movement.
mouseY	(Numeric/Dynamic) Map image pixel vertical coordinate of mouse pointer. 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 map varies 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.
shapeSelectBuffer	(Boolean/Dynamic) If true and value of shapeBufferDistance is greater than zero, a buffer of shapeBufferDistance will be applied in the spatial filter of a selection by shape.
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.

aimsCommon.js

chkGeocodeLayers	(Boolean/Dynamic) If true, the viewer will request a list of layers that have geocoding parameters configured. Default value is false. This variable tells whether or not a GET_SERVICE_INFO request should be sent to obtain layer information.
chkRouteLayers	(Boolean/Dynamic) If true, the viewer will request a list of layers that have routing parameters configured. This requires the Route Server extension.
chkUnits	(Boolean/Dynamic) If true, the viewer will do a simple test to see if the extent is not within practical limits for decimal degrees. In the test, if the coordinate values are too large, the map units are set to "FEET". Normally, the map units are set within Designer, and chkUnits is set to false.
coordsDelimiter	(String/Dynamic) Character to be used in delimiting x and y coordinates in ArcXML requests. Default character is a space (" ").
decimalChar	(String/Dynamic) Character to be used in decimal numbers by the browser. Viewer sets value at startup. Default character is a point (".").
isArcMapService	(Boolean/Static) If true, the service in the main map is an ArcMap Service. The value is set by the viewer on load of service.
legendImage	(String/Dynamic) String containing URL of image to be used in graphic legend.
legendTemp	(String/Dynamic) String used to temporarily contain URL of image to be used in graphic legend.
modeBlurb	(String/Dynamic) String containing current viewer mode to be displayed in ModeFrame.
ovIsVisible	(Boolean/Dynamic) If true, the overview map will be visible.
pairsDelimiter	(String/Dynamic) Character to be used in delimiting x and y coordinates in ArcXML requests. Default character is a semicolon (";").
processStartExtent	Checks if aimsRoutePresent is true. If true, and if useGeocode, useReverseGeocode, or useRoute is true, sends a GET_SERVICE_INFO request to Route Server for layer information.
showBuffer	(Boolean/Dynamic) If true, buffer instructions will be added to map request. Set to true by viewer when buffer tool is enabled.

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, GClablel, 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.
GClablel	(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 service.
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 service.
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 service.
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 service.
GCpointCount	(Numeric/Dynamic) Number of matching locations returned in geocode response. Updated by viewer with each geocode response.
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 the geocode request. The elements in this array have corresponding elements in the GCid, GCdesc, and GClabel arrays. Updated by user on Locate Address form.
geocodeAppMode	(String/Dynamic) Mode used for display of geocoding or routing responses. Set to <code>locate</code> by default. Other modes are <code>route</code> and <code>address</code> and are available only with the Route Server extension.
parseGeocodeLayers	Checks GET_SERVICE_INFO response to see if any of the name attributes within the GCSTYLE tags match the value of the variable SdcGeocodeStyle. If no match is found, it sets the values of the variables <code>useRoute</code> and <code>useReverseGeocode</code> to false, disabling routing and reverse geocoding.
parseGeocodeResults	Extended function that supports the Route Server extension. The value of the variable <code>geocode AppMode</code> is used to determine what will run at the end of this function.
SdcGeocodeStyle	(String/Static) Style of geocoding returned in geocode parameter response if Route Server extension is installed. If this style is not present, routing and reverse geocoding are disabled.

aimsIdentify.js

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.
AliasFieldAliases	(Array/Dynamic) Array of strings containing the alias of the alias field for each layer in the service. The elements of this array have corresponding elements in the AliasFieldName array. Updated by viewer on load service.
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 AliasFieldAlias array. Updated by viewer on load service.
canQuery	(Boolean/Dynamic) If true, selection and query can proceed. Updated by viewer on startup.
canSelectInvisible	(Boolean/Static) If true, invisible features can be selected. The default value is false, meaning invisible features can't be selected. This does not check for validity of the layer.
currentHyperLinkField	(String/Dynamic) Field name of current HyperLinkLayer field. Updated by viewer on start of Hyperlink mode.
currentHyperLinkLayer	(String/Dynamic) Name of current HyperLinkLayer. Updated by viewer on start of HyperLink mode.
currentHyperLinkPrefix	(String/Dynamic) Prefix to be used with current HyperLinkLayer field value. Updated by viewer on start of HyperLink mode.
currentHyperLinkSuffix	(String/Dynamic) Suffix to be used with current HyperLinkLayer field value. Updated by viewer on start of HyperLink mode.
fID	(Numeric/Dynamic) Number indicating layer index. Used by IdentifyAll and HyperLinkAny modes.
fIndex	(Numeric/Dynamic) Number indicating layer index. Used by HyperLinkAny mode.
hyperLinkWindowHeight	(Numeric/Static) Height of window used to display hyperlinked URL.
hyperLinkWindowWidth	(Numeric/Static) Width of window used to display hyperlinked URL.
idEast	(Numeric/Dynamic) Current extent right (maximum x coordinate). Used by IdentifyAll and HyperLinkAny modes.
idNorth	(Numeric/Dynamic) Current extent left (maximum y coordinate). Used by IdentifyAll and HyperLinkAny modes.
idSouth	(Numeric/Dynamic) Current extent bottom (minimum y coordinate). Used by IdentifyAll and HyperLinkAny modes.
idWest	(Numeric/Dynamic) Current extent left (minimum x coordinate). Used by IdentifyAll and HyperLinkAny modes.

newSelectCount	(Numeric/Dynamic) Number of current selected features. Updated by viewer with each query or selection.
queryStartRecord	(Numeric/Dynamic) Starting record position for returned records. Updated by viewer on each query or selection.
replyArray	(Array/Dynamic) Array containing ArcXML response strings. Used by IdentifyAll mode.
searchTolerance	(Numeric/Dynamic) Search tolerance. This value is calculated using pixelTolerance and extent-to-pixel ratio. Updated by viewer on identify and hyperlink requests.
selectPoints	(Array/Dynamic) Array of feature IDs for selected features. Updated by viewer with each query or selection on load service.

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 service. Updated by viewer on load of service.
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.
getLayers	Checks if layer information has been read in from a file. If not, process GET_SERVICE_INFO response for layer information.
layerCount	(Numeric/Dynamic) Number of layers in service. Updated by viewer on load service.
LayerExtent	(Array/Dynamic) Array of strings containing the extent for each layer in the service. 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 service.
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 service. The elements in this array have corresponding elements in the LayerExtent, LayerFieldPrecisionList, LayerFieldSizeList, LayerFieldTypeList, LayerID, LayerIDField, LayerIsFeature, LayerMaxScale, LayerMinScale, LayerName, LayerRenderString, LayerShapeField, LayerType, and LayerVisible arrays. Updated by viewer on load service.
LayerFieldPrecisionList	(Array/Dynamic) Array of strings containing lists of precisions for the fields of each layer in the service. 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 service.
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.
LayerFieldSizeList	(Array/Dynamic) Array of strings containing lists of sizes for the fields of each layer in the service. The elements in this array have corresponding elements in the LayerExtent, LayerFieldList, LayerFieldPrecisionList, LayerFieldTypeList, LayerID, LayerIDField, LayerIsFeature, LayerMaxScale, LayerMinScale, LayerName,

	LayerRenderString, LayerShapeField, LayerType, and LayerVisible. Updated by viewer on load service.
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 service. The elements in this array have corresponding elements in the LayerExtent, LayerFieldList, LayerFieldPrecisionList, LayerFieldSizeList, LayerID, LayerIDField, LayerIsFeature, LayerMaxScale, LayerMinScale, LayerName, LayerRenderString, LayerShapeField, LayerType, and LayerVisible arrays. Updated by viewer on load service.
LayerID	(Array/Dynamic) Array of strings containing the ID value for each layer in the service. The elements in this array have corresponding elements in the LayerExtent, LayerFieldList, LayerFieldPrecisionList, LayerFieldSizeList, LayerFieldTypeList, LayerIDField, LayerIsFeature, LayerMaxScale, LayerMinScale, LayerName, LayerRenderString, LayerShapeField, LayerType, and LayerVisible arrays. Updated by viewer on load service.
LayerIDField	(Array/Dynamic) Array of strings containing the names of ID fields for each layer in the service. The elements in this array have corresponding elements in the LayerExtent, LayerFieldList, LayerFieldPrecisionList, LayerFieldSizeList, LayerFieldTypeList, LayerID, LayerIsFeature, LayerMaxScale, LayerMinScale, LayerName, LayerRenderString, LayerShapeField, LayerType, and LayerVisible arrays. Updated by viewer on load service.
LayerIsFeature	(Array/Dynamic) Array of Boolean values for each layer in the service 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, LayerShapeField, LayerType, and LayerVisible arrays. Updated by viewer on load service.
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 clicking Hide LayerList or Display LayerList buttons.
LayerMaxScale	(Array/Dynamic) Array of strings containing the maximum scale threshold for each layer in the service. 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 service.
LayerMinScale	(Array/Dynamic) Array of strings containing the minimum scale threshold for each layer in the service.

LayerName	(Array/Dynamic) Array of strings containing the names of the layers in the service.
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 service.
LayerRenderString	(Array/Dynamic) Array of strings containing the render information for each layer in the service.
LayerShapeField	(Array/Dynamic) Array of strings containing the shape field for each layer in the service.
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.
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.
setLayerVisible	(Array/Dynamic) Array of boolean values indicating starting visibility of layers. This array may or may not be the same size of LayerVisible, but the index of each element in setLayerVisible matches the corresponding index in LayerVisible. Updated by getCommandLineParams in aimsMap.js if the viewer calling URL contains the argument Layers=xxxx..., where x is 1 or 0. Details on URL parameters are found in the commented lines of getCommandLineParams function.

aimsLegend.js`defaultLegTitle`

(String/Dynamic) String containing default legend title.

`drawLegendOnly`

(Boolean/Dynamic) If true and legendVisible is true, the current map image request will have the following tag:

```
<Draw map="false" />
```

This tag is used to designate that only an image for the legend be returned. No map image will be sent.

aimsMap.js

appDir	(String/Static) String containing URL path of viewer files. Set by viewer on startup.
canLoad	(Boolean/Dynamic) If true, viewer can load services interactively. Updated by viewer on startup. Available for custom implementation.
debugOn	(Numeric/Dynamic) Current debug mode. Available modes are: 0=Off, 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.
eBottom	(Numeric/Dynamic) Current extent bottom (minimum y coordinate). Formerly called bottom.
eLeft	(Numeric/Dynamic) Current extent left (minimum x coordinate). Formerly called left.
enforceFullExtent	(Boolean/Static) If true, map extent sent in map request will always be within full extent. Set by viewer on startup.
eRight	(Numeric/Dynamic) Current extent right (maximum x coordinate). Formerly called right.
eTop	(Numeric/Dynamic) Current extent top (maximum y coordinate). Formerly called top.
fullBottom	(Numeric/Dynamic) Full extent bottom (minimum y coordinate). Updated by viewer on load of service.
fullHeight	(Numeric/Dynamic) Full extent height. Updated by viewer on load of service.
fullLeft	(Numeric/Dynamic) Full extent left (minimum x coordinate). Updated by viewer on load of service.
fullOVBottom	(Numeric/Dynamic) Overview map full extent bottom (minimum y coordinate). Updated by viewer on load of service.
fullOVHeight	(Numeric/Dynamic) Overview map full extent height. Updated by viewer on load of service.
fullOVLeft	(Numeric/Dynamic) Overview map full extent left (minimum x coordinate). Updated by viewer on load of service.
fullOVRight	(Numeric/Dynamic) Overview map full extent right (maximum x coordinate). Updated by viewer on load of service.
fullOVTop	(Numeric/Dynamic) Overview map full extent top (maximum y coordinate). Updated by viewer on load of service.
fullOVWidth	(Numeric/Dynamic) Overview map full extent width. Updated by viewer on load of service.
fullRight	(Numeric/Dynamic) Full extent right (maximum x coordinate). Updated by viewer on load of service.

fullTop	(Numeric/Dynamic) Full extent top (maximum y coordinate). Updated by viewer on load of service.
fullWidth	(Numeric/Dynamic) Full extent width. Updated by viewer on load of service.
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 service.
imageLimitLeft	(Numeric/Dynamic) Limit extent left (minimum x coordinate) of defined image size. Updated by viewer on load of service.
imageLimitRight	(Numeric/Dynamic) Limit extent right (maximum x coordinate) of defined image size. Updated by viewer on load of service.
imageLimitTop	(Numeric/Dynamic) Limit extent top (maximum x coordinate) of defined image size. Updated by viewer on load of service.
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.

legendVisible	(Boolean/Dynamic) If true, map request will include request for legend image. Updated by user by clicking the 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 or 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.
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 or 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 or response.
queryZoom	(Boolean/Dynamic) If true and a SQL query expression has been passed in the loading URL, the viewer will zoom to around the selected feature on startup.
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 Internet Explorer. Updated by viewer as needed.
toolMode	(Numeric/Dynamic) Cursor mode. Updated by user on click of toolbar buttons.
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 or response.
yDistance	(Numeric/Dynamic) Current map extent height. Updated by viewer on each map request or response.

aimsMap.js (defined values for toolMode)

toolMode	Description
1	Zoom In
2	Zoom Out
3	Pan
4	Identify
5	Identify All
8	Query/Find/Search
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
30	HyperLink Any
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.

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 or variables used for user input values in each StoredQuery available for the current active layer. Updated by viewer on start of StoredQuery mode.

aimsResource.js

buttonList	(Array/Static) Array of strings containing text to be displayed on buttons.
modeList	(Array/Static) Array of strings containing text to be displayed indicating the current tool in the viewer.
msgList	(Array/Static) Array of strings containing text to be displayed in the viewer.
sUnitList	(Array/Static) Array of strings containing text to be used to set unit types in requests to the server.
titleList	(Array/Suite) Array of strings containing text to be displayed in titles.
unitList	(Array/Suite) Array of strings containing text to be displayed for unit types.

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 hyperlink in display of returned record values of selected features or through custom implementation.
queryMode	(Numeric/Dynamic) Current type of query or selection process: zero=spatial selection, 1=attribute query. Updated by viewer on start of appropriate query or selection mode.
selectBottom	(Array/Dynamic) Array of numbers representing the bottom coordinate of the extent of each selected feature. Updated by viewer on query or 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 or 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 on start of appropriate query or selection mode.
selectLayer	(String/Dynamic) ID of the layer that query or 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 or selection response.
selectRight	(Array/Dynamic) Array of numbers representing the right coordinate of the extent of each selected feature. Updated by viewer on query or selection response.
selectTop	(Array/Dynamic) Array of numbers representing the top coordinate of the extent of each selected feature. Updated by viewer on query or selection response.
selectType	(String/Dynamic) The shape type of the layer that query or selection will be issued against. Updated by viewer as needed.
selMaxEnvelope	(Array/Dynamic) Array used to contain the minimum and maximum x coordinates and minimum and maximum y coordinates of the returned selected features. These values are calculated in the function calcSelectEnvelope() in aimsSelect.js.

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 or selection request.
shapeBufferDistance	(Numeric/Dynamic) Current buffer distance to be used in Select by Line or Polygon or Select by Shape mode. Only used if shapeSelectBuffer is set to true.

aimsXML.js

charEncoding	(String/Dynamic) String containing locale encoding for ArcXML requests. Default is “UTF-8”.
charSet	(String/Dynamic) String containing charset value for dynamic pages written by the viewer. Default value is “ISO-8859-1”. Used for internationalization or location of dynamic pages.
connectorURL	(String/Static) String containing the URL to the “home” Servlet Connector. This is automatically set on load by the viewer with the Web page’s host and servlet path. The Web pages must originate from the same Web server used by the Application Server to avoid JavaScript permission errors.
cVersion	(String/Static) String containing version of client. Used in requests to server.
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.
forceNewOVMap	(Boolean/Dynamic) If true, a new overview map image will be requested with each new map image. Default is false, where the overview map image is requested only on startup. The extent box is updated by the viewer.
formColor	(String/Dynamic) Color of the “hidden” form page in PostFrame expressed in either HTML hexadecimal RGB format (“#rrggbb”) or standard HTML color name, for example, “white”, “purple”, or “red”.
hyperlinkXMLMode	(Numeric/Static) Value to be used for HyperLink requests in XMLMode.
identifyXMLMode	(Numeric/Static) Value to be used for Identify requests in XMLMode.
localeEncoding	(String/Dynamic) String containing complete locale encoding string for ArcXML requests. Uses value of charEncoding. Default is ‘encoding= “UTF-8”’.
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 service. Updated on first map request.
queryXMLMode	(Numeric/Static) Value to be used for query requests in XMLMode.
requestMethod	(String/Static) Mode used for which Connector the ArcXML request will be directed to. The choices are Servlet or Java, with Servlet as the default. The value of this variable will be set to the value of connectorType in the parent frame page viewer.htm.
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.
writeXML	Checks for insertion of appropriate tags for displaying of routes in requests for map images. These tags are written by functions provided in Route Server extension samples.

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) Global variable whose number represents the current ArcXML request or response mode. The mode determines which function to call. 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
5	List of Services
6	Graphic Selection
7	Identify
8	Query/Find/Search
9	Geocode
10	Identify All
11	Buffer
14	Find
15	HyperLink
16	Hyperlink Any
20	Startup Query
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 Database
98	Map and 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
902	Overview Map Image Only
> 1,000	Custom Function (useCustomFunction is called)

ArcIMSParam.js

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

hostName	(String/Dynamic) Name of machine where Web document originated. Set by the user on startup.
esriBlurb	(String/Static) String used in portion of request URL. By default this is “/servlet/com.esri.esrimap.Esrimap?ServiceName=”. This references the path to the Servlet Connector and preceeds the Service Name in the URL.
catURL	(String/Dynamic) URL to request catalog of services using hostName. Set by the user on startup.
serverURL	(String/Dynamic) String containing service URL prefix used in loading service Images.

The following variables are written by ArcIMS Designer.

imsURL	(String/Dynamic) URL of Image Service used in main map display.
imsOVURL	(String/Dynamic) URL of Image Service used in overview map Display. If value is “” (empty quotes), value of imsURL is used here.
imsQueryURL	(String/Dynamic) URL of service used for querying. If value is “” (empty quotes), value of imsURL is used here appended with “&CustomService=Query”.
imsGeocodeURL	(String/Dynamic) URL of service 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 RGB value. If value is “”(empty quotes), color defined in map configuration file is used.
ovBoxColor	(String/Dynamic) Color of Extent Box in overview map expressed as an RGB 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.

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 the service configuration 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 service configuration 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.

The following set of variables beginning with “use” are all (Boolean/Dynamic). Each sets whether or not the tool is displayed on the 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 on the toolbar.
usePanNorth	If true, displays panNorth button on the toolbar.
usePanWest	If true, displays panWest button on the toolbar.
usePanEast	If true, displays panEast button on the toolbar.
usePanSouth	If true, displays panSouth button on the toolbar.
useZoomIn	If true, displays ZoomIn button on the toolbar.
useZoomOut	If true, displays ZoomOut button on the toolbar.
useFullExtent	If true, displays FullExtent button on the toolbar.
useZoomActive	If true, displays ZoomActive button on the toolbar.
useZoomLast	If true, displays ZoomLast button on the toolbar.
useIdentify	If true, displays Identify button on the toolbar.
useMeasure	If true, displays Measure button on the toolbar.
useSetUnits	If true, displays SetUnits button on the toolbar.
useSelect	If true, displays Select button on the toolbar.
useQuery	If true, displays Query button iLayerID array on the toolbar.
useFind	If true, displays Find button on the toolbar.
useGeocode	If true, displays Geocode button on the toolbar.
	If you have licensed the RouteServer extension and if aimsRoutePresent in MapFrame.htm is true, you can set this to true to have the viewer send a _SERVICE_INFO request to the RouteServer for layer information.

useStoredQuery	If true, displays Stored Query button on the toolbar.
useClearSelect	If true, displays Clear Select button on the toolbar.
usePrint	If true, displays Print button on the toolbar.
useBuffer	If true, displays Buffer button on the toolbar.
useExtract	If true, displays Extract button on the toolbar. Extract currently not implemented.

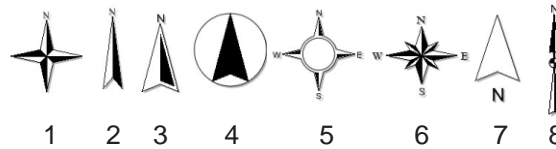
Additional tool settings:

useHyperLink	(Boolean/Dynamic) If true, this function is available, and a button will be displayed on the default toolbar.
useHyperLinkAny	(Boolean/Dynamic) If true, this function is available, and a button will be displayed on the default toolbar.
useIdentifyAll	(Boolean/Dynamic) If true, this function is available, and a button will be displayed on the default toolbar.
useBufferShape	(Boolean/Dynamic) If true, this function is available, and a button will be displayed on the default toolbar.
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.

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 is 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.
showScalePercent	(Boolean/Dynamic) If true, displays 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.
doURLencode	(Boolean/Dynamic) If true, the ArcXML responses will be URL encoded. Default is false.
autoAdjustForArcMapServer	(Boolean/Static) If true, viewer will adjust parameters for ArcMap Server.

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 counterclockwise direction. A value of 0 is pointing right or “East”, 90 is pointing up or “North”.
 drawScaleBar (Boolean/Static) If true, a scalebar 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 can 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 RGB value.
 ScaleBarFontColor (String/Static) Color of the text font expressed as an RGB value.
 ScaleBarColor (String/Static) Color of the bar expressed as an RGB 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.
 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.
 drawScaleBar2 (Boolean/Static) If true, the optional second scalebar is drawn on the map.
 ScaleBar2Units (String/Dynamic) Type of units for ScaleBar2, buffer, and measure display. Possible values are FEET, METERS, MILES, or KILOMETERS. Can be updated by user if useSetUnits is true.
 ScaleBar2Background (Boolean/Static) If true, text has a background.
 ScaleBar2BackColor (String/Static) Color of the text background expressed as an RGB value.

ScaleBar2FontColor	(String/Static) Color of the text font expressed as an RGB value.
ScaleBar2Color	(String/Static) Color of the bar expressed as an RGB value.
ScaleBar2Font	(String/Static) The text font face. This value is case sensitive.
ScaleBar2Style	(String/Static) The text font style.
ScaleBar2Round	(String/Static) Number of digits to round.
ScaleBar2Size	(String/Static) Length of bar in pixels.
ScaleBar2Width	(String/Static) Width of bar in pixels.
ScaleBar2Precision	(String/Static) Number of decimal places.
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 RGB 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 RGB value.
CopyrightGlow	(Boolean/Static) If true, text has a glow effect.
CopyrightGlowColor	(String/Static) Color of the glow expressed as an RGB value.
CopyrightShadow	(Boolean/Static) If true, text has shadow effect.
CopyrightShadowColor	(String/Static) Color of the shadow expressed as an RGB value.
CopyrightText	(String/Static) Text string.
drawBottomBar	(Boolean/Static) If true, a background bar is underneath the copyright text and scalebar.
bottomBarColor	(String/Static) Color of the bottom bar expressed as an RGB value.
bottomBarOutline	(String/Static) Color of the outline of the bottom bar expressed as an RGB value.
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 the next request for a new map image.
modeMapColor	(String/Static) Color of the mode text expressed as an RGB value.
modeMapGlow	(String/Static) Color of the glow expressed as an RGB 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.
mapTransparent	(Boolean/Dynamic) If true, the map image background will be transparent. This requires GIF or PNG output of map image.
transColor	(String/Dynamic) Color to be used for transparency of map background expressed as an RGB value. A value is required. Used if mapTransparent is true.
is5up	(Boolean/Dynamic) If true, a version 5, or newer, browser is being used.
isIE4	(Boolean/Dynamic) If true, Microsoft Internet Explorer version 4 is being used.
isNav4	(Boolean/Dynamic) If true, Netscape Navigator version 4 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:	
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 a 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.
useExternalWindow	(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 a 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.
hideLayersFromList	(Boolean/Dynamic) If true, use the noListLayer array for a list of layers to not list in the LayerList or Legend. Default is false.
noListLayer	(Array/Dynamic) Array of boolean values (true or false) designating if a layer should be listed in the LayerList or Legend. An element must be defined for each layer, which is the zero element representing the top layer. This array is only referenced if the value of hideLayersFromList is true.
drawFloatingMode	(Boolean/Static) If true, current tool mode will be drawn on a style sheet (Netscape layer) in a MapFrame (DHTML display).
modeLayerOn	(Boolean/Dynamic) If true, current status of displaying Tool mode on a style sheet (Netscape layer) in a MapFrame (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 a 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 point markers at the click location when Measure, Shape Selection, or buffer is performed.

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

Identify/Select/Query/Buffer parameters:

pixelTolerance	(Numeric/Dynamic) Search tolerance in pixels around click.
selectColor	(String/Dynamic) Color for Selected Features expressed as an RGB value.
highlightColor	(String/Dynamic) Color for Highlighted Feature expressed as an RGB value.
transparentLevel	(String/Dynamic) Transparency level of fill color used to display selected and highlighted features.
zoomToSingleSelect	(Boolean/Dynamic) If true and only one feature is returned in a query or selection, then viewer will zoom in to area surrounding selected feature.
onlyUniqueSamples	(Boolean/Static) If true, use only unique values in sample field value lists.
queryCaseInsensitive	(Boolean/Static) If true, queries are case insensitive. Default is false, so HTML Viewer matches Java Viewers.

Note: Field names must be uppercase.

selectFields (String/Dynamic) A list of fields to be returned in an identify, selection, or query request... #ALL#=all fields. If not #ALL#, then ID and shape fields are required in list.

For example, 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.

selFieldList (Array/Dynamic) Array of strings containing a list of fields to be returned in query or 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 a 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 headings.

fieldAliasList	<p>(Array/Dynamic) Array of strings containing a list of field names and the aliases to be used as column headings in the display of field values returned in a query or 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.</p> <pre>fieldAliasList [0]= ""; fieldAliasList [1]= "NAME:City Name POP:Population"; fieldAliasList [2]= ""; fieldAliasList [3]= ""; fieldAliasList [4]= "";</pre> <p>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 headings. 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 (:).</p>
hideIDFieldData	<p>(Boolean/Dynamic) If true, the ID field will not be shown in the display of query response. Default is false.</p>
hideShapeFieldData	<p>(Boolean/Dynamic) If true, the shape field will not be shown in the display of query response. Default is false.</p>
hyperLinkLayers	<p>(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.</p>
hyperLinkFields	<p>(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.</p> <pre>hyperLinkLayers[0] = "Image"; hyperLinkFields[0] = "URL";</pre> <p>Sample set. In this case, the layer 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.</p>
hyperLinkPrefix	<p>(Array/Dynamic) Array of strings containing the string that should be put in front of the value of the corresponding element in the hyperLinkFields array to make an appropriate URL. Each element must have a corresponding element in the hyperLink Layers and hyperLinkFields arrays. If no prefix is to be used, an empty string ("") should be used for the value.</p>
hyperLinkSuffix	<p>(Array/Dynamic) Array of strings containing the string that should be appended to the value of the corresponding element in the hyperLinkFields array to make an appropriate URL. Each element must have a corresponding element in the hyperLinkLayers and hyperLinkFields arrays. If no suffix is to be used, an empty string ("") should be used for the value.</p>

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 or selection.
listAllLayersInIDAll	(Boolean/Dynamic) If true, all layers will be listed in returned IdentifyAll response display, even if no features were found for some layers. Default is false, listing only the layers with features found at click location.
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.
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 a 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 a 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.
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.

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 RGB 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. If you have licensed the RouteServer extension and if aimsRoutePresent in MapFrame.htm is true, you can set this to true to have the viewer send a GET_SERVICE_INFO request to the RouteServer for layer information.
useRoute	(Boolean/Static) If true, routing functionalities are available. Default value is false. Used in the same fashion as the rest of the “use<tool>” variables, enabling the use of the tool. The Route Server extension must be installed. If you have licensed the RouteServer extension and if aimsRoutePresent in MapFrame.htm is true, you can set this to true to have the viewer send a GET_SERVICE_INFO request to the RouteServer for layer information.

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.
aimsRoutePresent	<p>(Boolean) Is set to true after the aimsRoute.js file has been loaded. Default value is false. The RouteServer extension must be installed to use the functions in this .js file. This file, containing JavaScript functions for communicating with the RouteServer, is included in the RouteServer extension's sample HTML Viewer. Additional files for forms and results are also included.</p> <p>If this field is set to true, and if useGeocode, useReverseGeocode, or useRoute is true, a GET_SERVICE_INFO request is sent to the RouteServer for layer information.</p>
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) Offset 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.
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.

