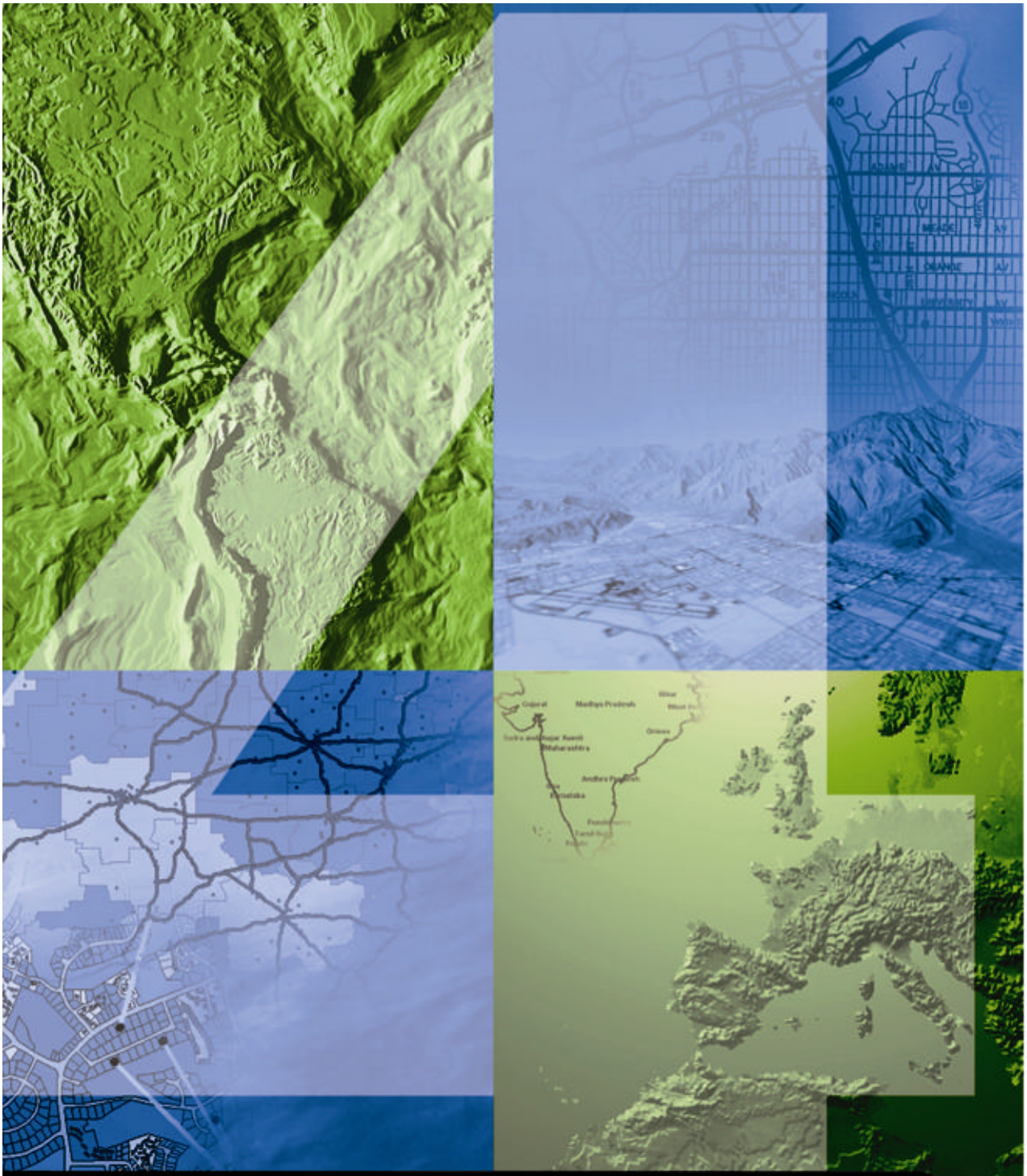


ArcXML Programmer's Reference Guide

ArcIMS® 4

ESRI ArcGIS™

Copyright © 2000–2002 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 the copyright laws of the given countries of origin and applicable international laws, treaties, and/or conventions. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying 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, ArcSDE, and the ArcGIS logo are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions. Microsoft and the Windows logo are registered trademarks and the Microsoft Internet Explorer logo is a trademark of Microsoft Corporation.

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

Table of Contents

ArcXML Conventions 1
Additional ArcXML Documents 4
What's New in ArcXML 1.1 for ArcIMS 4.0 5
Introduction to ArcXML 10
Using Map and Viewer Configuration Files 17
    Comparing Different CONFIG Files 45
Using ArcXML Renderers 47
    Introduction 47
    A Case Study 69
Using Projection Elements 80
    Introduction 80
    Projection Elements in ArcIMS 82
    Using the Projection Elements 84
    Defining a Coordinate System 92
    Using Datum Transformations 97
    Data Densification 98
    Processing Time 99
Using GET_SERVICE_INFO and SERVICEINFO with Image and Feature Services 100
    Introduction 100
    GET_SERVICE_INFO Framework 102
    SERVICEINFO Framework 102
    GET_SERVICE_INFO Attributes 106
Using GET_SERVICE_INFO and SERVICEINFO with ArcMap Image Services 116
    Introduction 116
    GET_SERVICE_INFO Framework 117
    SERVICEINFO Framework 117
    GET_SERVICE_INFO Attributes 121
Using GET_IMAGE and IMAGE with Image Services 132
    Introduction 132
    GET_IMAGE Request and IMAGE Response 134
    Zoom and Pan with ENVELOPE 137
    Changing the Output IMAGESIZE 137
    Using BACKGROUND 142
    LAYERLIST and LAYERDEF 143
    Adding Dynamic Layers 150
    Using Projections with GET_IMAGE 159
    Using LEGEND and DRAW 162
    Using OUTPUT to Control Image Names and Locations 172
Using GET_IMAGE and IMAGE with ArcMap Image Services 177
    Introduction 177
    GET_IMAGE Request and IMAGE Response 177
    Zoom and Pan with ENVELOPE 180
    Changing the Output IMAGESIZE 181
    Using BACKGROUND 184

Using LAYERLIST and LAYERDEF 185
Adding Information in a Request 188
Using Projections with GET_IMAGE 194
Using LEGEND and DRAW 197
Using OUTPUT to Control Image Names and Locations 200
Comparison of Elements and Attributes used with the Image and ArcMap Servers 204
Using GET_LAYOUT and LAYOUT with ArcMap Image Services 250
Introduction 250
GET_LAYOUT Request and LAYOUT Response 250
Using ENVELOPE 253
Using SCALE 254
Changing the Layout Format and Output Size using OUTPUT, LAYOUT, GET_LAYOUT autoresize 255
Using OUTPUT to Control Layout Names and Locations 257
Using Projections with GET_LAYOUT 261
Using GET_EXTRACT and EXTRACT 266
Introduction 266
Creating a Map Configuration File with an Extract Extension 266
Routing to the Extract Server 270
Using the GET_EXTRACT Request 270
The EXTRACT Response 271
Customizing the GET_EXTRACT Request 271
Transforming Geometry with the Extract Server 279
Using GET_FEATURES and FEATURES 281
Introduction 281
Accessing an ArcIMS Service using GET_FEATURES 283
Outputmode 285
Geometry and Compact 286
Beginrecord, Featurelimit, Count, Hasmore 288
Skipfeatures 290
Envelope, Attributes, and Checkesc 292
Globalenvelope 293
Transforming Geometry 294
Summary of Geocoding Elements 297
Introduction 297
Geocode Elements Used in a Map Configuration File 297
Geocode Elements Used in GET_GEOCODE Requests 298
Geocode Elements Used in a GEOCODE Response 300
Geocode Elements Used in a SERVICEINFO Response 300
Using Custom Address Styles 303
Introduction 303
What Are Address Styles? 303
Implementing a Custom Address Style 304
Elements of the ArcIMS Style File 305
Example ArcIMS Style File 316
ADD_RELEVANCE_FEEDBACK 321
ADDEDFEATURES 323

ADDRESS 325
ADMIN_TABLE 326
ADMINCMD 328
ARCXML 330
AVIMSWORKSPACE 332
BACKGROUND 334
BAND 336
BANDS 337
BUFFER 339
CALLOUTMARKERSYMBOL 345
CAPABILITIES 348
CHANGE_METADATA_ACCESS 350
CHARTSYMBOL 352
CHARTVALUE 358
CLEANUP 364
COLLECTION_INFO 366
CONFIG 368
CONTENT_INFO 371
COORDS 373
COORDSYS 376
DATAFRAME 381
DATAFRAMEINFO 383
DATASET 385
DELETE_METADATA 393
DELETE_METADATA_RELATIONSHIP 395
DELETEDFEATURES 397
DENSIFY 399
DOCUMENTINFO 401
DRAW 402
ENVELOPE 404
ENVIRONMENT 409
ERROR 411
EXACT 413
EXTENSION 418
EXTRACT 421
EXTRACTPARAMS 422
FCLASS 424
FEATURE 426
FEATURECOORDSYS 429
FEATURECOUNT 435
FEATURES 437
FEATURESERVERWORKSPACE 440
FIELD 442
FIELDS 448
FIELDVALUE 450
FILTERCOORDSYS 453
FULLTEXT 457

GCCOUNT....459
GCFIELD....461
GCINPUT....463
GCSTYLE....465
GCTAG....471
GEOCODE....473
GET_COLLECTION_INFO....475
GET_CONTENT_INFO....477
GET_EXTRACT....478
GET_FEATURES....480
GET_GEOCODE....484
GET_IMAGE....486
GET_LAYOUT....490
GET_METADATA....492
GET_METADATA_DOCUMENT....493
GET_RASTER_INFO....495
GET_ROOT_DATASET....497
GET_SERVICE_INFO....498
GET_USER....501
GET_UUID....502
GETCLIENTSERVICES....503
GRADIENTFILLSYMBOL....504
GROUPRENDERER....506
HASHLINESYMBOL....508
HOLE....511
IMAGE....513
IMAGEGENERALIZATION....515
IMAGELIMIT....518
IMAGEPROPERTIES....520
IMAGESERVERWORKSPACE....523
IMAGESIZE....525
IMAGEWORKSPACE....528
LAYER....532
LAYERDEF....540
LAYERINFO....544
LAYERLIST....547
LAYERS....551
LAYOUT....554
LAYOUTINFO....555
LEGEND....557
LINE....565
LOCALE....568
MAP....570
MAPUNITS....572
MARKUP....575
MARKUPLAYER....577
METADATA....579

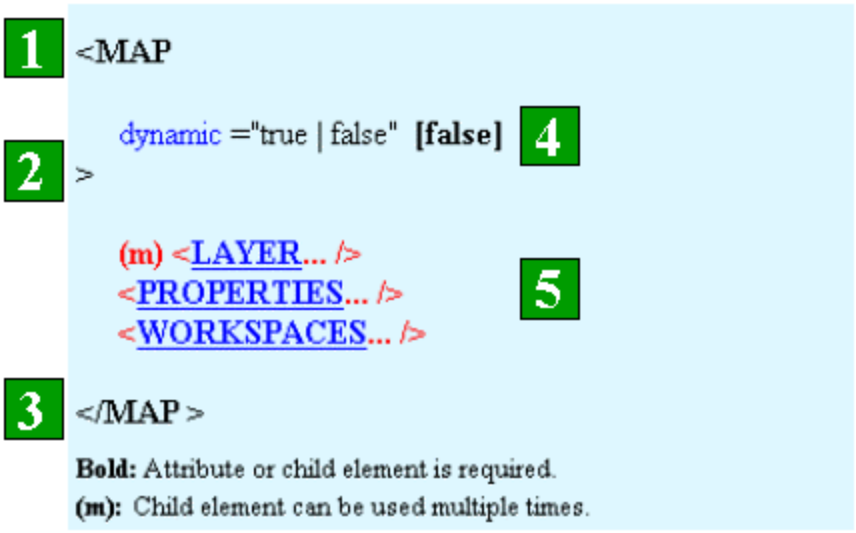
METADATA_ACTION....581
METADATA_CHILD....582
METADATA_CONFIG....584
METADATA_CONTENT....585
METADATA_DATASET....587
METADATA_SIBLING....590
METADATA_SOURCE....592
MODIFIEDFEATURES....594
MOIMSWORKSPACE....596
MULTIPOINT....598
NORTHARROW....602
OBJECT....605
OTHER....614
OUTPUT....617
OUTPUTFIELD....624
OUTPUTFILE....626
OVERVIEWMAP....628
PARTITION....630
PATH....634
POINT....637
POLYGON....643
POLYLINE....648
PROPERTIES....651
PUBLISH_METADATA....655
PUT_METADATA....657
PUT_METADATA_RELATIONSHIP....660
PUT_METADATA_SEMANTIC....661
PUT_USER....662
QUERY....663
RANGE....667
RASTER_EXACT....671
RASTER_INFO....673
RASTER_OTHER....674
RASTER_RANGE....676
RASTER_RENDERER....678
RASTERFILLSYMBOL....680
RASTERMARKERSYMBOL....683
RASTERSHIELDSYMBOL....687
RENAME_METADATA....692
REQUEST....694
RESET....696
RESPONSE....698
RESPONSE_COLUMN....700
RING....702
SCALE....705
SCALEBAR....707
SCALEDEPENDENTRENDERER....711

SCREEN 716
SDEWORKSPACE 718
SEARCH_METADATA 722
SEMANTIC_PAIR 726
SEPARATORS 728
SERVICE 732
SERVICEINFO 737
SERVICES 743
SHAPEWORKSPACE 745
SHIELDSYMBOL 747
SIMPLELABELRENDERER 750
SIMPLELINESYMBOL 757
SIMPLEMARKERSYMBOL 761
SIMPLEPOLYGONSYMBOL 764
SIMPLERENDERER 767
SPATIALFILTER 769
SPATIALQUERY 772
SQVAR 783
STOREDQUERIES 785
STOREDQUERY 787
SUBSET 789
TABLE_NAME 791
TAG 793
TAGTEXT 795
TAGVALUE 797
TARGETLAYER 799
TEXT 801
TEXTMARKERSYMBOL 804
TEXTSYMBOL 807
THUMBNAIL 810
TOC 811
TOCCLASS 813
TOCGROUP 815
TRUETYPEMARKERSYMBOL 817
UIFONT 823
UPDATED 825
USER 826
UUID 827
VALUEMAPLABELRENDERER 828
VALUEMAPRENDERER 835
WORD 839
WORKSPACES 840

ArcXML Conventions

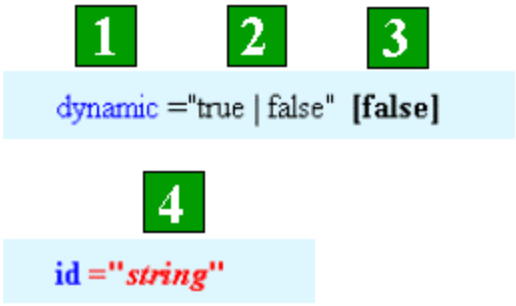
The ArcXML element structure includes the element name, attributes, and child elements.

Element Name



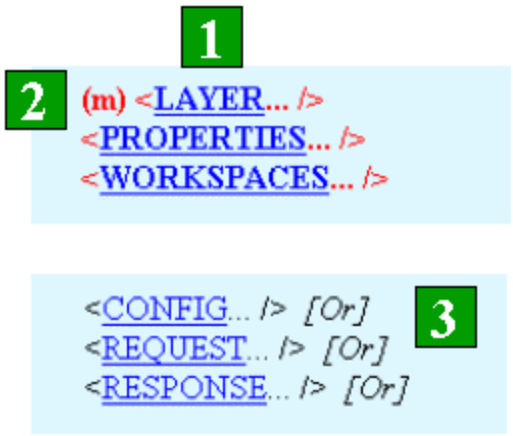
1. The opening element starts with a less than symbol (<) and is followed by its name. Elements are always written in uppercase letters.
2. The opening element ends with a greater than symbol (>).
3. The closing element is similar to the opening element. It begins with the less than symbol and a forward slash (/) and is followed by its name. The element ends with a greater than symbol.
4. If the element includes any attributes, they are included after the opening element.
5. All child elements are inserted before the closing element.

Attributes



1. Attributes are always written in lowercase letters. If the attribute is required, it is bold. A definition of each attribute is found in the attribute table for the element.
2. If an attribute has a defined list of valid values, all possible values are listed. Only one value can be used at a time. Attribute values are always placed inside of double quotes ("), and the text is not case sensitive unless noted.
3. If the attribute has a default value, it is listed in brackets after the attribute list or type.
4. If an attribute does not have a list of known values, the value type is listed such as double, integer, and string.

Child Elements



1. Child elements are always written in uppercase letters. If the child element is required, it is bold. Each child element is linked to the corresponding page describing that element.
2. Some child elements can be used multiple times. If this is the case, the letter "m" in parentheses (m) is in front of the child element.

3. In some cases, special instructions are given in brackets after the element. The most common scenario is when a group of child elements is listed but only one child element can be used. In this example, [Or] means to select one child element from the group. More details are given in the Restrictions section to explain special instructions.

Additional ArcXML Documents

In addition to this version of the *ArcXML Programmer's Reference Guide* in PDF format, the following related documents and files are also included with ArcIMS:

- An ArcXML DTD. An XML Data Type Definition (DTD) is available for use with ArcXML 1.1. This DTD defines the structure rules for the elements and attributes in ArcXML and checks for document validity. The DTD can be used with many XML editors, but since ArcIMS requires UTF8 encoding, editors that support UTF8 encoding are recommended. The following two editors support UTF8 encoding:

  - XML Spy: http://www.xmlspy.com
  - Xeena: http://alphaworks.ibm.com/tech/xeena

  A note when using ArcIMS Author in files edited with an XML editor: in some XML editors, when a DTD is used, a DOCTYPE statement is written at the top of the document. The DOCTYPE contains information on the location of the DTD. If you open a map configuration file in ArcIMS Author that contains a DOCTYPE statement, this line will be deleted if you save the file in ArcIMS Author.

- The ArcXML Programmer's Reference Guide in WebHelp.

- A directory of shapefiles commonly used with examples throughout the guide. Shapefiles included are cntry94, cities, states, province, world30, and some New York City shapefiles.

On Windows, the ArcXML documents and files are installed during a typical setup and are located in the following directory:

- <ArcIMS Installation Directory>\Documentation\ArcXML_Guide

The files are also available from the ArcIMS Custom Setup. On the Installation Type screen, select Custom. On the next panel, the ArcIMS Components and Installation Directory screen, uncheck all components already checked, and check Documentation. Next, click Options and select only the *ArcXML Programmer's Reference Guide*. All the ArcXML documents are copied to the same ArcXML_Guide directory as listed above.

On UNIX, the DTD, the *ArcXML Programmer's Reference Guide* in WebHelp and PDF formats, and the shapefiles are found in the directory:

- $AIMSHOME/Documentation/ArcXML_Guide

The ArcXML documents are also available from ArcIMS Online: http://arcimsonline.esri.com.

What's New in ArcXML 1.1 for ArcIMS 4.0

ArcXML 1.1 has several new elements and attributes. The new or changed features are described below. New elements were added to support the Metadata and ArcMap Servers. In addition, some elements were added for supporting some new functionality with raster images.

New Documents and Examples

Several documents have been added to the ArcXML 1.1 introductory section that provide comprehensive explanations and additional examples on the ArcMap Server:

- Using GET_IMAGE and IMAGE with ArcMap Image Services
- Using GET_LAYOUT and LAYOUT with ArcMap Image Services
- Using GET_SERVICE_INFO and SERVICEINFO with ArcMap Image Services

New Elements to support the Metadata Server

The following elements have been added to support the Metadata Server. For more information on using the Metadata Server, see *Creating and Using Metadata Services*.

These elements are used to publish metadata documents to the Metadata Server.

- CHANGE_METADATA_ACCESS
- DELETE_METADATA
- DELETE_METADATA_RELATIONSHIP
- GET_UUID
- METADATA_ACTION
- METADATA_CHILD
- METADATA_SIBLING
- METADATA_SOURCE
- PUBLISH_METADATA
- PUT_METADATA
- PUT_METADATA_RELATIONSHIP
- PUT_METADATA_SEMANTIC
- PUT_USER
- RENAME_METADATA
- RESET
- SEMANTIC_PAIR
- THUMBNAIL
- UUID

These elements are used to browse metadata documents that have been published to the Metadata Server.

- ADD_RELEVANCE_FEEDBACK
- COLLECTION_INFO
- CONTENT_INFO
- DOCUMENTINFO
- FULLTEXT
- GET_COLLECTION_INFO
- GET_CONTENT_INFO
- GET_METADATA
- GET_METADATA_DOCUMENT
- GET_ROOT_DATASET
- GET_USER
- METADATA
- METADATA_ACTION
- METADATA_DATASET
- SEARCH_METADATA
- SUBSET
- TAG
- TAGTEXT
- TAGVALUE
- UPDATED
- USER
- WORD
- ENVELOPE has a new attribute ***spatialoperator*** to support metadata searches.

These elements are used in metadata configuration files.

- ADMIN_TABLE
- METADATA_CONFIG
- METADATA_CONTENT
- RESPONSE_COLUMN
- TABLE_NAME

New Elements to support the ArcMap Server

The following elements have been added to support the ArcMap Server. Additional information on using the ArcMap Server elements can be found in the introductory section of this guide.

- DATAFRAME
- DATAFRAMEINFO
- GET_LAYOUT
- LAYOUT
- LAYOUTINFO

- SCALE
- TOC
- TOCCLASS
- TOCGROUP

The following elements have new attributes to support the ArcMap Server.

- GET_FEATURES.
  - *dataframe*
- GET_IMAGE
  - *dataframe*
- GET_SERVICE_INFO
  - *dataframe*
  - *toc*
  - *toctype*

New Elements to support raster images (Image Server only)

The following elements can be used in map configuration files to classify image pixel values.

- RASTER_EXACT
- RASTER_OTHER
- RASTER_RANGE
- RASTER_RENDERER

The following element can be used to turn off image generalization.

- IMAGEGENERALIZATION

New Elements to support identifying raster pixel values (Image and ArcMap Servers)

The following elements can be used to identify the value of a pixel in an Image or ArcMap Image Service.

- BAND
- BANDS
- GET_RASTER_INFO
- RASTER_INFO

Elements with New Attributes

The following elements have new attributes. Only the new attributes are listed here:

- BUFFER
  - *project*

Deprecated Elements

The following elements have been deprecated in ArcIMS 4.0. You can still use them, but ArcIMS 4.0 is the last version they will be supported.

- AVIMSWORKSPACE
- MOIMSWORKSPACE

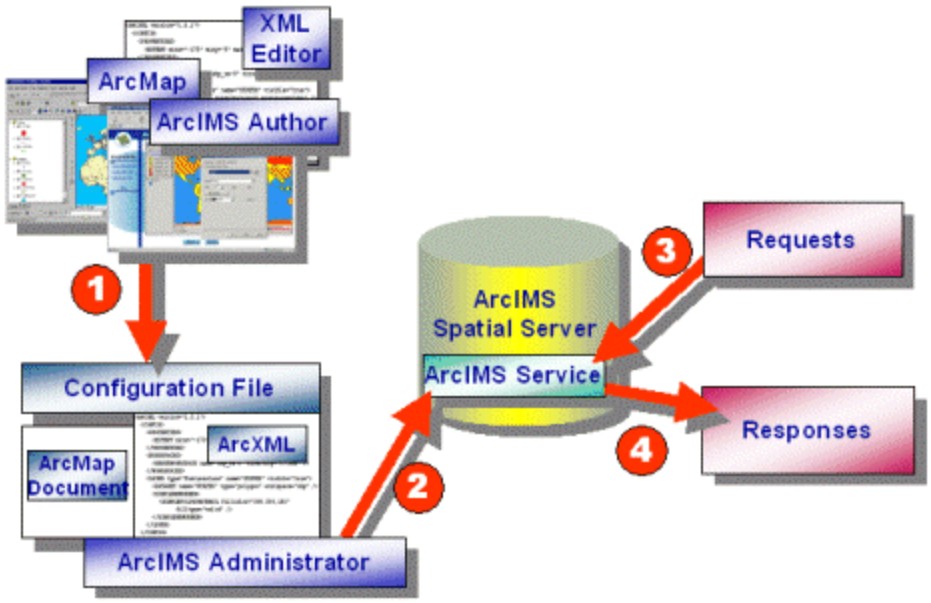
Section 1: Introduction to ArcXML

Introduction to ArcXML

ArcXML is the protocol for communicating with the ArcIMS Spatial Server. An ArcIMS Spatial Server is the backbone of ArcIMS and provides the functional capabilities for accessing and bundling maps and data into the appropriate format before sending the data back to a client. In order to understand ArcXML, it is first necessary to understand how configuration files, ArcIMS services, requests, and responses relate to each other and how they interact with the ArcIMS Spatial Server.

The figure below is a diagram showing the interaction between the ArcIMS Spatial Server and configuration files, services, requests, and responses. The numbers in the diagram reflect the different steps involved in communicating with the ArcIMS Spatial Server:

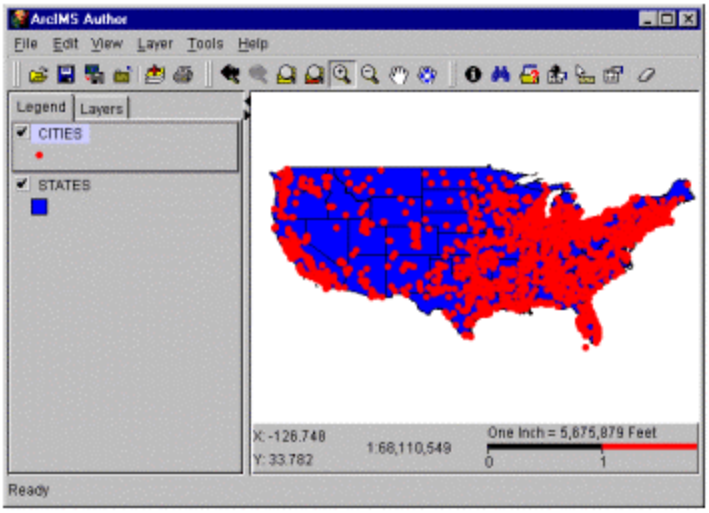
- In Step 1, for Image and Feature Services, you create a ***map configuration file***. For Metadata Services, you create a ***metadata configuration file***. The map and metadata configuration files are written in ArcXML. For ArcMap Image Services, you use ArcMap to generate ***ArcMap documents***.
- In Step 2, you use ArcIMS Administrator to start an ***ArcIMS Service*** on the ArcIMS Spatial Server. The configuration file from Step 1 is the input to the service.
- In Step 3, the ArcIMS Spatial Server receives a ***request*** in ArcXML.
- In Step 4, the ArcIMS Spatial Server generates a ***response*** in ArcXML.



Step 1: Creating a Configuration File

In Step 1, a configuration file is generated. You can create a map configuration file for Image and Feature Services using ArcIMS Author or by using a text or XML editor. For Metadata Services, you create a metadata configuration file by using a text or XML editor. For ArcMap Image Services, you use ArcMap to generate an ArcMap document, which is treated as a map configuration file by ArcIMS. These files are in binary format and cannot be edited outside of ArcMap.

The next figure shows a map in ArcIMS Author with two layers: STATES and CITIES.



When this file is saved in ArcIMS Author, the result is a map configuration file, written in ArcXML, containing layer information on STATES and CITIES. The file has an *.axl extension. In this example, assume the file is named *usa.axl*.

Usa.axl map configuration file using ArcXML:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-125" miny="25" maxx="-67" maxy="50"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="C:\ESRIDATA\USA" />
      </WORKSPACES>
      <LAYER type="featureclass" name="States" visible="true"
id="States">
        <DATASET name="STATES" type="polygon" workspace="shp_ws-0" />
```

```
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="0,0,255" filltype="solid" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Cities" visible="true"
id="Cities">
        <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="255,0,0" width="6" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

The element that distinguishes a map configuration file from a request or response is CONFIG. Elements within the CONFIG element, such as PROPERTIES, WORKSPACES, and LAYER, help define the characteristics of the map.

Step 2: Starting an ArcIMS Service

An ArcIMS service is a process that runs on the ArcIMS Spatial Server. You can think of a service as a portal to the Spatial Server. Spatial Server functionality is accessible only through services running on the server.

In Step 2, a map configuration file, such as usa.axl or an ArcMap document (*.mxd or *.pmf), is the input to an ArcIMS service. When starting a service, you also must assign the service to a Feature Server, Image Server, or ArcMap Server. In each case, a map configuration file provides drawing instructions for each layer in a service. These instructions are the default state for the service.

The naming of an ArcIMS service is independent of the name of the input map configuration file. For example, usa.axl can be the input file to an Image Service named *usa_image*. This service is instructed, by default, to draw the STATES layer with a blue polygon fill and the CITIES layer with a red marker.

Step 3: Sending a Request

In Step 3, once an ArcIMS service such as usa_image is running on the ArcIMS Spatial Server, requests can be sent to the service. Requests are generated by the ArcIMS HTML Viewer, Java Viewers, and any viewers using the ColdFusion, ActiveX, and Java Connectors. The requests are:

- GET_IMAGE
- GET_FEATURES
- GET_GEOCODE
- GET_EXTRACT
- GET_SERVICE_INFO
- GET_RASTER_INFO

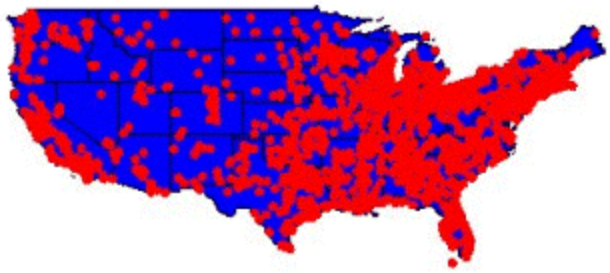
- GET_LAYOUT
- GET_METADATA
- PUBLISH_METADATA

The element that distinguishes a request from other types of ArcXML files is REQUEST. For example, to request a map, a GET_IMAGE request can be made:

A sample request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-125" miny="25" maxx="-67" maxy="50" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

When the above request is sent to the usa_image service, the result is a map that looks like the original map defined in ArcIMS Author. The layer symbol definitions from the service are used: blue for the states and red for the cities.



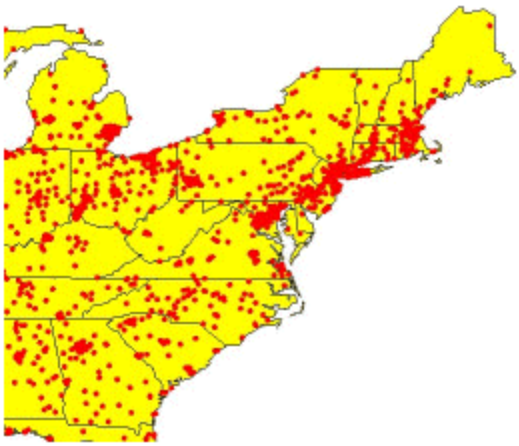
A request can also override some of the information in a service by asking for a new map at a different scale, turning layers on and off, requesting a subset of the attribute data, changing the projection, or adding acetate layers, among other things. With the Image Server, a request can also be used to change the rendering of a layer or add new data in dynamic layers. In the next example to an Image Service, a request is sent to the usa_image service asking for information at a new zoom scale and changing the color of the States layer from blue to yellow.

Request to change the zoom area and layer color of an ArcIMS service:

```
<?xml version="1.0"?>
<ARCXML version="1.0">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-88" miny="30" maxx="-67" maxy="50.0" />
        <IMAGESIZE width="500" height="350" />
        <LAYERLIST>
          <LAYERDEF id="States" visible="true" >
```

```
            <SIMPLERENDERER>
              <SIMPLEPOLYGONSYMBOL filltype="solid"
fillcolor="255,255,0" />
            </SIMPLERENDERER>
          </LAYERDEF>
        </LAYERLIST>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The new map, when drawn, has new rendering for the States layer at a zoomed in scale.



Note that because of architectural differences with the ArcMap Server, layer rendering and symbology cannot be changed through a request to an ArcIMS Image Service. Using the same request above, the same extent would show, but the rendering for the States layer would not change from what is in the ArcMap document.

Step 4: Receiving a Response

In Step 4, when the ArcIMS Spatial Server processes a request, the results are returned in a response. The element that distinguishes a response from other types of ArcXML files is RESPONSE. For example, an IMAGE response contains the name and location of the map generated by the ArcIMS Spatial Server. The following example shows a possible IMAGE response from the usa_image service.

A sample response:

```
<?xml version="1.0"?>
<ARCXML version="1.0">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-87.5" miny="30.0" maxx="-59.5" maxy="50.0" />
      <OUTPUT file="C:\ArcIMS\output\usa_image_MYCOMPUTER2953026.jpg"
url="http://mycomputer.domain.com/output/usa_image_MYCOMPUTER2953026.jpg"
/>
    </IMAGE>
```

```
  </RESPONSE>
</ARCXML>
```

There are nine responses in ArcIMS that correspond to the requests as shown in the following table.

Request	Response	More Information
GET_IMAGE	IMAGE	• Using GET_IMAGE and IMAGE with Image Services • Using GET_IMAGE and IMAGE with ArcMap Image Services
GET_FEATURES	FEATURES	• Using GET_FEATURES and FEATURES
GET_GEOCODE	GEOCODE	• Summary of Geocoding Elements
GET_EXTRACT	EXTRACT	• Using GET_EXTRACT and EXTRACT
GET_SERVICE_INFO	SERVICEINFO	• Using GET_SERVICE_INFO and SERVICEINFO with Image and Feature Services • Using GET_SERVICE_INFO and SERVICEINFO with ArcMap Image Services
GET_LAYOUT	LAYOUT	• Using GET_LAYOUT and LAYOUT

		with ArcMap Image Services
GET_RASTER_INFO	RASTER_INFO	
GET_METADATA	METADATA	
PUBLISH_METADATA	Refer to child elements of PUBLISH_METADATA	

Using Map and Viewer Configuration Files

Introduction

Configuration files are used to define a map. There are four types of configuration files:

- A **map configuration file used with Image and Feature Servers.** This file is created using ArcIMS Author or an XML editor and is used as input to an ArcIMS service. The output file is in ArcXML.
- A **map configuration file used with ArcMap Server.** This file is created using ArcMap. The output files are ArcMap documents (*.mxd and *.pmf).
- A **viewer configuration file.** This file is generated when a map in ArcExplorer 4 or an ArcIMS Java Viewer is saved.
- **Default.axl.** This file is a special type of viewer configuration file that is generated during the ArcIMS Designer process and is used as input to the ArcIMS Java Viewers.

The following sections cover configuration files in detail. The first section discusses map configuration files for Image and Feature Services, the next section covers viewer configuration files, and the third section reviews default.axl. A final section provides a summary review of the similarities and differences among the configuration file types.

Note: Metadata configuration files and map configuration files created using ArcMap are not discussed in this document.

Map Configuration Files

Map configuration files are used as input to ArcIMS services. All information in a map configuration file provides a default set of instructions for map properties and rendering. Requests can override information in the service properties, but if the request does not include any special instructions, then the default service properties prevail. The diagram below shows the flow for creating an ArcIMS service using a map configuration file as input.

- In Step 1, data sources are referenced and rendered. This process takes place using ArcIMS Author or by using a text or XML editor for Image and Feature Services. If non-English characters are used for layer names or font names, either ArcIMS Author or an XML editor must be used. Text editors do not display non-English characters correctly. For ArcMap Image Services, map configuration files are generated using ArcMap.
- In Step 2, when a session is saved in ArcIMS Author, an XML editor, or ArcMap, the output is a map configuration file.

Tool	File Format	File Extension	Valid ArcIMS Services
Author	ArcXML file	*.axl	Feature and Image Services
XML Editor	ArcXML file	*.axl	Feature and Image Services
ArcMap	ArcMap map file	*.mxd	ArcMap Image Services
ArcMap	Published map file	*.pmf	ArcMap Image Services

Note that data sources are not included in the file, only references to the data. The information in a map configuration file provides a default set of instructions for map properties and rendering.

- In Step 3, the map configuration file is used as input in ArcIMS Administrator to start an Image, Feature, or ArcMap Image Service.
- In Step 4, once the service has been started it runs as a process on the ArcIMS Spatial Server and is ready to receive requests.



Framework of a map configuration file

This next section focuses on the framework of ArcXML map configuration files used as input to Image and Feature MapServices. An ArcXML map configuration file can be divided into several sections. The following example shows the basic framework.

Framework of a map configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>...</ENVIRONMENT>
    <MAP>
      <PROPERTIES>...</PROPERTIES>
      <WORKSPACES>...</WORKSPACES>
      <LAYER>...</LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

A map configuration file includes:

- A prolog, which is used to define the XML version and encoding.
- An ARCXML element, which is used in all ArcXML statements.
- A CONFIG element, which is used in all configuration files.
- An ENVIRONMENT element, which is used to define the locale.
- A MAP element, which is used to define the map.

ARCXML and the prolog

The first line of an ArcXML statement is the prolog. All ArcXML 1.1 statements are required to use a standard prolog that includes the XML version and encoding. The XML version is 1.0. Encoding in ArcXML 1.1 is UTF8.

After the prolog, all ArcXML statements begin and end with ARCXML. This element is required, and the version number is 1.1.

Prolog and ARCXML:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
...
</ARCXML>
```

CONFIG, ENVIRONMENT, and MAP

A map configuration file distinguishes itself from a REQUEST or RESPONSE by the CONFIG element. For more information on the relationship between map configuration files, requests, and responses, see Introduction to ArcXML.

The only child elements to CONFIG in a map configuration file are ENVIRONMENT and MAP.

CONFIG, ENVIRONMENT, and MAP:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>...</ENVIRONMENT>
    <MAP>...</MAP>
  </CONFIG>
</ARCXML>
```

ENVIRONMENT is used to set up information about the environment used in the ArcIMS service. Once the environment is established, clients in one locale can access a service created in another locale. Child elements of ENVIRONMENT include the following:

- LOCALE. (Required) LOCALE is used to set the country and language for the locale of the ArcIMS site. This information is based on settings in the operating system. For example, in the United States, the country is "US" and the language is "en". Some characteristics of the United States locale include using English alphabetic characters and a decimal point (.) to represent number fractions such as 123.456.

  In Germany, the country is "DE" and the language is "de". Some characteristics of the German locale include using non-English characters and a comma (,) to represent number fractions such as 123,456.

- UIFONT. (Required) UIFONT is used to set a default font for the dialogs in ArcExplorer 4 and the ArcIMS Java Viewers.
- SEPARATORS. (Optional) SEPARATORS is used to denote characters to separate x,y coordinates and coordinate pairs. By default, x,y coordinates are separated by a space and coordinate pairs are separated by a semicolon (;). These can be changed using SEPARATORS. The attribute *cs* is the coordinate separator and is used to separate an x-coordinate from a y-coordinate. The attribute *ts* is the tuple separator and is used to separate x,y coordinate pairs.
- SCREEN. (Optional) SCREEN is used to set the resolution of the screen for the computer generating the map configuration file. This value is important for computing scale dependencies. By default, the dots per inch (dpi) is assumed to be 96 dpi. This is the value for most machines with a display size set to 1024 x 768 pixels.

ENVIRONMENT and its child elements:

```
<ENVIRONMENT>
  <LOCALE language="en" country="US" />
  <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
  <SEPARATORS cs=" " ts=";"/>
  <SCREEN dpi="96"/>
</ENVIRONMENT>
```

The element MAP contains all the instructions for generating a map. This includes:

- PROPERTIES. The PROPERTIES section includes the initial map extent, map units, and current projection, as well as additional instructions used for Image Services.
- WORKSPACES. The WORKSPACES section includes the location of all the data used to create map layers.
- LAYER. One LAYER element is used for each layer in a map. LAYER contains the information about how the data should be symbolized.

MAP and its child elements:

```
<MAP>
  <PROPERTIES>...</PROPERTIES>
  <WORKSPACES>...</WORKSPACES>
  <LAYER>...</LAYER>
</MAP>
```

The PROPERTIES, WORKSPACES, and LAYER elements are discussed in detail in the next sections.

MAP: PROPERTIES

PROPERTIES provides the framework for defining properties about an ArcIMS service. The child element ENVELOPE is required but all other child elements are optional. The child elements shown in the following example are used in both Image and Feature Services.

PROPERTIES and its child elements for both Feature and Image Services:

```
<PROPERTIES>
  <ENVELOPE minx="-105.594842" miny="-49.955227" maxx="75.672764"
maxy="83.596039" name="Initial_Extent" />
  <MAPUNITS units="decimal_degrees" />
  <FEATURECOORDSYS id="54008" />
  <FILTERCOORDSYS id="54008" />
</PROPERTIES>
```

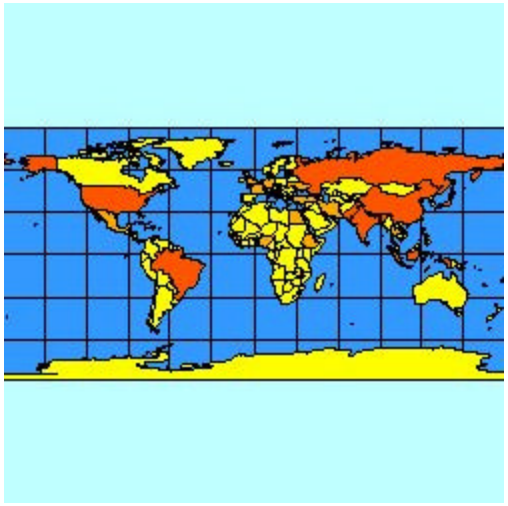
- ENVELOPE. In a map configuration file, ENVELOPE defines the initial extent for display. The initial extent is made up of a rectangle bounded by minimum and maximum x,y coordinates. In the next example, the initial extent uses decimal degree coordinates.

  Using ENVELOPE:

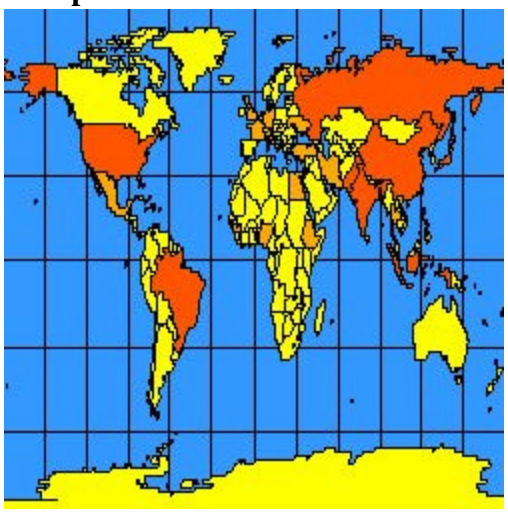
  ```
  <PROPERTIES>
    <ENVELOPE minx="-61.1" miny="3.7" maxx="91.7" maxy="61.3"
  name="Initial_Extent" reaspect="true" />
    ...
  <PROPERTIES>
  ```

Note the attribute *reaspect* was included in the above example. This attribute indicates whether the ENVELOPE should be stretched to fit the viewing area in the client. By default, *reaspect* is set to "true" and the pixel width and height ratio stays the same. By setting *reaspect* to "false", the pixel width and height are stretched to fit the viewing area. The following two figures show the difference when *reaspect* is used. The figure on the left has *reaspect* set to true. The ENVELOPE is recalculated so that the map does not stretch. The figure on the right has *reaspect* set to "false", and the map has been stretched to fit the viewing area.

Reaspect set to true



Reaspect set to false



- MAPUNITS. MAPUNITS defines the units of the data used in the map. Units are decimal degrees, feet, or meters. If the elements FEATURECOORDSYS and FILTERCOORDSYS are not present, the author of the map configuration file is responsible for determining the MAPUNITS. In this case, if MAPUNITS is not included, the assumption is that the MAPUNITS are in decimal degrees. When MAPUNITS is incorrectly specified, measurements, buffers, and other functions using scales will not be correct, and the requested map may not contain any data. If FEATURECOORDSYS and FILTERCOORDSYS are present, the correct MAPUNITS is automatically selected by the ArcIMS Spatial Server. If MAPUNITS is included in the map configuration file, it is ignored.

- FEATURECOORDSYS and FILTERCOORDSYS. FEATURECOORDSYS and FILTERCOORDSYS set the projection of the service. For an overview of the projection elements, see Using Projection Elements.

The following child elements are valid only with Image Services and are more commonly used in requests than in a map configuration file:

PROPERTIES and its child elements valid only with Image Services (in bold):

```
<PROPERTIES>
  <ENVELOPE minx="-105.594842" miny="-49.955227" maxx="75.672764"
maxy="83.596039" name="Initial_Extent" />
  <MAPUNITS units="decimal_degrees" />
  <FEATURECOORDSYS id="54008" />
  <FILTERCOORDSYS id="54008" />
  <BACKGROUND... />
  <LEGEND... />
  <OUTPUT... />
</PROPERTIES>
```

- BACKGROUND. BACKGROUND defines a color for the image background. It can also be used to make one color in the image transparent. Depending on the browser, the image formats that support transparent colors vary. JPG images do not support transparent colors. The table below lists which image formats support transparent colors for different browsers.

Browser	Supported Transparent Image Formats
ArcIMS HTML Viewer in Internet Explorer	PNG8, GIF
ArcIMS HTML Viewer in Netscape 4.x	GIF
ArcIMS HTML Viewer in Netscape 6.x	PNG8, GIF
ArcExplorer 4	PNG24, PNG8, GIF
ArcIMS Java Viewers in Internet Explorer and Netscape	PNG24, PNG8, GIF

  To make a color transparent, both the *color* and *transcolor* attributes of BACKGROUND must be set to the same color. When a color is set to transparent, layers underneath the transparent part of the image can be seen.

  When the ArcIMS HTML Viewer is used, BACKGROUND added to an Image Service is overridden by BACKGROUND in requests from the HTML Viewer. When the ArcIMS Java Viewers and ArcExplorer 4 are used, the BACKGROUND in the service is not overridden.

- LEGEND. LEGEND defines a map's legend for an Image Service. This legend is valid only in customized ArcIMS HTML Viewers and not in the ArcIMS Java Viewers or ArcExplorer 4. When LEGEND is included in the map configuration file, a legend image is always generated. However, this legend is ignored by the ArcIMS HTML Viewer. The HTML Viewer generates a legend upon request and ignores any other generated legends. For more information on using LEGEND, see the Using LEGEND and Draw Section in Using GET_IMAGE and IMAGE.

- OUTPUT. OUTPUT defines a location and filename for the output maps and legend image files. If OUTPUT is not defined, the image output directory and URL location are determined at the time the ArcIMS service is started, and the ArcIMS Spatial Server assigns a filename. The ArcIMS Tasker, a Windows service or UNIX daemon, automatically deletes these generated images on a user-specified interval.

  When OUTPUT is used in a map configuration file, the OUTPUT information overrides the output information stored when the service is started. When OUTPUT is used, the output files are not automatically deleted by ArcIMS Tasker. In order for the files to be deleted, the *taskfile* property must be set in tasker.properties. For information on setting this property, see the topic "Setting up ArcIMS Monitor and ArcIMS Tasker Property Files" in the System Administration: Administering Property Files chapter of the ArcIMS Help Contents.

  When using an ArcIMS HTML Viewer, ArcIMS Java Viewer, ArcExplorer 4, or any other browser using the ArcIMS Servlet Connector, the OUTPUT element is restricted by default. In a map configuration file, OUTPUT is ignored. This restriction can be lifted by setting the property *spatialServer.AllowRequestOutput* to true in esrimap_prop. This property file is found in the servlet engine directory. For more information on the location of esrimap_prop and its properties, see the topic "Administering the ArcIMS Servlet Connector" in the ArcIMS Help Contents.

  These restrictions apply only when the ArcIMS Servlet Connector is used. They do not apply to the ActiveX or ColdFusion Connectors or to other connectors using the ArcIMS Application Server Link. The OUTPUT filename extensions are restricted to *.jpg, *.png, or *.gif, regardless of whether OUTPUT is restricted or not.

  In general, it is not recommended to use OUTPUT in a map configuration file unless all requests can use the same output map or legend. Instead, if OUTPUT is needed, it should be used in GET_IMAGE and GET_EXTRACT requests. During a request, more information is known about the requesting client, and the output location and name can be tailored for that client. For more details on using OUTPUT, see the Using OUTPUT to Control Image Name and Locations Section in Using GET_IMAGE and IMAGE.

MAP: WORKSPACES

WORKSPACES specifies the location of all the data used in the map configuration file. All data locations must be visible to any computer hosting ArcIMS services. Each workspace must have a unique reference name. Valid WORKSPACES in map configuration files are:

- SHAPEWORKSPACE. Used to reference directories with shapefiles.

- IMAGEWORKSPACE. Used to reference directories with images. Images can be accessed individually, as an image directory where all images in one directory can be tiled, or as image catalogs. A complete list of supported images is found in *Using ArcIMS*.
- SDEWORKSPACE. Used to reference layers in ArcSDE. The layers can be raster or vector layers. Either ArcSDE or ArcSDE Direct Connect can be used.

Valid WORKSPACES in a map configuration file:

```
<WORKSPACES>
  <SHAPEWORKSPACE name="shp_ws-0" directory="c:\ESRIDATA"/>
  <IMAGEWORKSPACE name="jai_ws-1" directory="c:\ESRIDATA"/>
  <SDEWORKSPACE name="sde_ws-2" server="zephyr" instance="port:5150"
database="" user="washoe" encrypted="true" password="LXEMUR" />
</WORKSPACES>
```

MAP: LAYER

LAYER is the parent element for defining a map layer. Attributes cannot be overridden by a request. The following attributes are required:

- *Id* is a unique ID for a layer. When using ArcIMS Author, layers are assigned a number. However, the ID can be any combination of alpha and numeric characters. Each layer must have a unique ID or an error will result when trying to start an ArcIMS service.
- *Type* specifies layer type. Use "featureclass" for shapefiles and ArcSDE vector layers. Use "image" for raster image files, GRIDs, and ArcSDE raster layers. Use "acetate" for adding graphics on top of the map.

The following attributes are optional:

- *Name* is used for the layer name displayed in the legend in all ArcIMS Viewers. If *name* is not included, the ArcIMS Java Viewers use the LAYER ID, and the HTML Viewer displays a blank name. It is highly recommended to always use *name*.

- *Maxscale* and *minscale* are used to set the maximum and minimum scales that the layer displays. A relative scale can be used, such as 1:24000, or the scale can be calculated as the number of map units per pixel. The method for calculating units per pixel is shown with the LAYER element information. If these attributes are not used, the layer displays at all scales.

  Applying maximum and minimum scales to a layer is a very important part of authoring map configuration files. The user should only see data that is relevant and useful at any given scale. If a layer has very detailed information, then this layer should not be available until the user has zoomed in to an appropriate level. The less data that must be accessed and rendered, the faster the map is drawn or data is transferred. A general guideline is that a map should contain no more than 500 or so features at a given scale. In some cases, there may be more features, but

realize that the amount of time to generate the map increases, especially when a map is projected.

- *Visible* specifies whether the layer is turned on or off in the legend when the map is first accessed. If the ArcIMS Java Custom Viewer is used, selecting which layers to have visible is done during the ArcIMS Designer process. This overrides *visible* in the ArcIMS service. All other ArcIMS Viewers read *visible* directly from the service.

Sample LAYER with attributes:

```
<LAYER type="featureclass" name="CITIES" minscale="1:24000"
maxscale="1:100000" visible="true" id="2"> ... </LAYER>
```

MAP: LAYER child elements

A LAYER used in a map configuration file has several child elements. The purpose of these elements is to identify the data source and to render the data.

LAYER with child elements:

```
<LAYER type="featureclass" name="CITIES" minscale="1:24000"
maxscale="1:100000" visible="true" id="2">
  <DATASET name="Cities_Robinson" type="polygon" workspace="shp_ws-0"
/>
  <DENSIFY tolerance="10000" />
  <COORDSYS id="54030" />
  <SPATIALQUERY where="population > 1000000" />
  <SIMPLERENDERER>
    <SIMPLEPOLYGONSYMBOL filltransparency="1.0" fillcolor="27,127,127"
/>
  </SIMPLERENDERER>
</LAYER>
```

The types of child elements can be divided into several groups: DATASET, projection elements, query elements, EXTENSION elements, and renderer and symbol elements.

- DATASET is required for a layer. It defines which data from a workspace to use. Information in DATASET cannot be overridden by a request.

  The attribute *name* is required. For shapefiles, use the name of the data file without an extension such as STATES. For ArcSDE, use the full name of the layer such as DATA.STATES. For images, the name depends on the method for accessing an image. For more details on naming images, see the table in the Notes section in DATASET.

  The attribute *type* is required for vector data and optional for image data. The value for *type* must match the shapefile or ArcSDE layer type: "point", "line", or "polygon". For images, always use the value "image".

The attribute *workspace* is required and references which workspace the data layer is in. The *workspace* value must be listed under WORKSPACES.

- The projection elements COORDSYS and DENSIFY are used if the data layer is in a different projection than the ArcIMS service. These elements are metadata stating what projection the layer is in, but they do not reproject the data. For more information on the projection elements, see Using the Projection Elements.

- The query elements SPATIALQUERY and QUERY are used to select a subset of the layer. SPATIALQUERY should be used for both tabular and spatial queries. QUERY should only be used in stored queries.

  When SPATIALQUERY is used with a layer in a map configuration file, a filter is set on a subset of the data and only that subset is available for viewing. When a request is made with SPATIALQUERY, the data outside of the ArcIMS service subset cannot be accessed.

- The EXTENSION element is used to define geocoding, stored queries, and extract parameters for a layer. All three extensions can be used for a single layer, but each EXTENSION can only have one set of child elements.

  For geocoding, use GCSTYLE. This extension sets up the type of geocoding that is permitted on the layer. More information on geocoding is available in Summary of Geocoding Elements.

  For setting up parameters for the Extract Server, use EXTRACTPARAMS. Using this extension allows HTML clients to send requests that extract specified layers of an ArcIMS service into a set of shapefiles, yielding one shapefile for each layer. More information on the Extract EXTENSION is available in Using GET_EXTRACT and EXTRACT.

  For setting up a stored query, use STOREDQUERIES. A stored query sets up a predefined query used in the ArcIMS Viewers.

- The renderer and symbol elements are used to define the symbology and labeling for each layer in the map. The relationship between a renderer and a symbol in ArcXML is

  ```
  <RENDERER>
    <SYMBOL />
  </RENDERER>
  ```

  Six renderers are available in ArcIMS:

  - SIMPLERENDERER
  - SIMPLELABELRENDERER
  - VALUEMAPRENDERER
  - VALUEMAPLABELRENDERER

- GROUPRENDERER
- SCALEDEPENDENTRENDERER

VALUEMAPRENDERER and VALUEMAPLABELRENDERER also have three child elements used to create value maps:

- RANGE
- EXACT
- OTHER

Only one symbol or label at a time can be used inside a renderer. However, the renderers can be used in different combinations to create complex symbols at different scales.

Valid symbols inside SIMPLERENDERER and VALUEMAPRENDERER are:

- **Point layers**
  - RASTERMARKERSYMBOL
  - SIMPLEMARKERSYMBOL
  - TRUETYPEMARKERSYMBOL
- **Line layers**
  - All the symbols valid for point layers
  - HASHLINESYMBOL
  - SIMPLELINESYMBOL
- **Polygon layers**
  - All the symbols valid for point and line layers
  - GRADIENTFILLSYMBOL
  - RASTERFILLSYMBOL
  - SIMPLEPOLYGONSYMBOL

Valid symbols inside SIMPLELABELRENDERER and VALUEMAPLABELRENDERER are:

- CALLOUTMARKERSYMBOL
- CHARTSYMBOL
- RASTERSHIELDSYMBOL
- SHIELDSYMBOL
- TEXTSYMBOL

For more detailed information on the renderers, see Using Renderer Elements.

Sample map configuration file

The following map configuration file uses many of the elements discussed above.

Example map configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-135.0" miny="-64.0" maxx="84.0" maxy="86.0"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-4" directory="<path to WORLD
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="Oceans" visible="true" id="0">
        <DATASET name="WORLD30" type="polygon" workspace="shp_ws-4" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="51,153,255"/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Countries" visible="true"
id="1">
        <DATASET name="country" type="polygon" workspace="shp_ws-4" />
        <GROUPRENDERER>
          <VALUEMAPRENDERER lookupfield="POP_CNTRY">
            <RANGE lower="0" upper="50000000" label="Less than
50000000">
              <SIMPLEPOLYGONSYMBOL fillcolor="255,255,0"/>
            </RANGE>
            <RANGE lower="50000001" upper="100000000" label="50000000 -
100000000">
              <SIMPLEPOLYGONSYMBOL fillcolor="255,170,0" />
            </RANGE>
            <RANGE lower="100000001" upper="1281008319" label="Greater
than 100000000">
              <SIMPLEPOLYGONSYMBOL fillcolor="255,85,0"/>
            </RANGE>
          </VALUEMAPRENDERER>
          <SCALEDEPENDENTRENDERER upper="1:35000000">
            <SIMPLELABELRENDERER field="CNTRY_NAME">
              <TEXTSYMBOL antialiasing="true" font="Arial Bold"
fontstyle="bold" fontsize="12" />
            </SIMPLELABELRENDERER>
          </SCALEDEPENDENTRENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

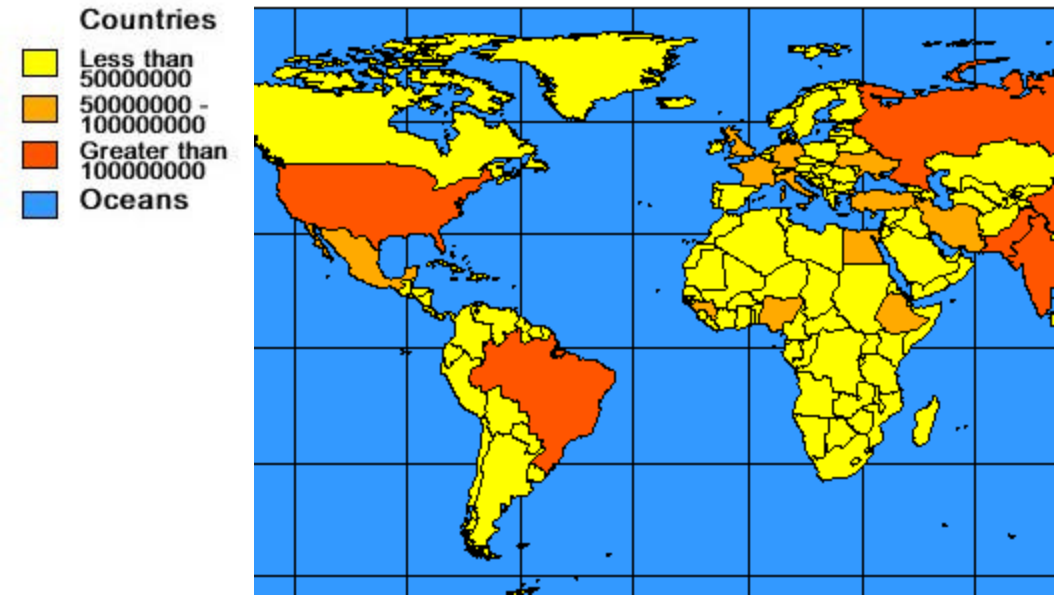
The map configuration file contains information on the locale, map properties,

workspaces, and how each layer should be drawn. In the ENVIRONMENT section, the LOCALE *country* and *language* are the "US" and "En". For UIFONT, the default *font* of Arial is used, and the default *dpi* in SCREEN is 96. The PROPERTIES elements show that the MAPUNITS are in decimal degrees and the ENVELOPE coordinates are -135.0,-64.0 and 84.0,86.0. In the WORKSPACES section, one SHAPEWORKSPACE is included for shapefiles in the WORLD ESRIDATA dataset.

The map contains two layers:

1. Ocean. This layer contains one renderer, SIMPLERENDERER. The layer has a solid blue fill as described in SIMPLEPOLYGONSYMBOL.
2. Countries. This layer contains several renderers. A VALUEMAPRENDERER is used to define how each country should be colored based on population. Countries with a population less than 50 million are yellow, those with a population between 50 and 100 million are light orange, and those with a population greater than 100 million are dark orange. A SCALEDEPENDENTRENDERER is used for labeling. The labels do not turn on until the scale is less than 1:35000000.

The following figure shows a map drawn based on the instructions in the map configuration file.



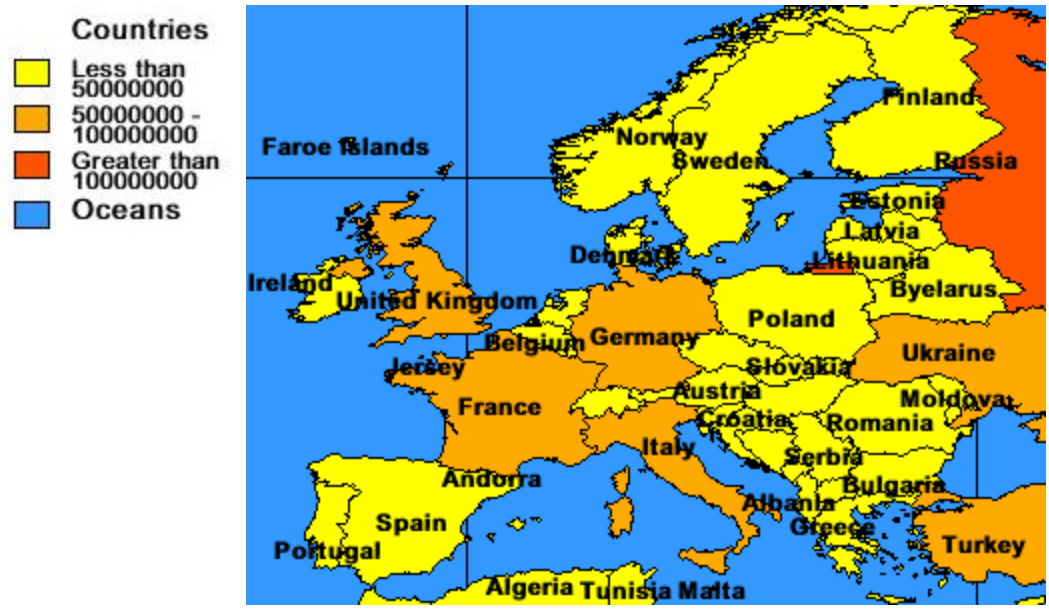
If the initial extent is changed so the scale is less than 1:35000000, the map zooms in enough so that labels for countries appear.

Change in initial ENVELOPE:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>...</ENVIRONMENT>
```

```
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-13.0" miny="36.0" maxx="34.0" maxy="69.0"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>...</WORKSPACES>
      <LAYER>...</LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

The following figure shows a map at a scale of about 1:35000000 and includes labels.



Viewer Configuration Files

Viewer configuration files are the output when a file is saved in ArcExplorer 4 or the ArcIMS Java Viewers. The diagram below provides a more detailed flow of the process to create a viewer configuration file:

- In Step 1, ArcIMS services and local data sources are accessed and viewed in ArcExplorer 4 or an ArcIMS Java Viewer.
- In Step 2, when a session is saved, the output is a viewer configuration file.



Framework of a viewer configuration file

The framework of a viewer configuration file is similar to a map configuration file. The framework includes the CONFIG, ENVIRONMENT, and MAP elements. These elements function the same in both map and viewer configuration files. PROPERTIES is fundamentally the same as well, but only ENVELOPE and MAPUNITS are included.

In many cases, a viewer configuration file can look just like a map configuration file. This is true when only local data sources are included in the file and the only WORKSPACES present are SHAPEWORKSPACE, IMAGEWORKSPACE, and SDEWORKSPACE.

Viewer configuration files can contain additional elements that are not in a map configuration file. The following example shows some of these elements.

Example viewer configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.0" miny="12.89" maxx="-68.0" maxy="83.59"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <IMAGESERVERWORKSPACE name="mapper_ws-6"
url="http://mymachine.domain.com/servlet/com.esri.esrimap.Esrimap"
service="background" />
        <FEATURESERVERWORKSPACE name="ifs_ws-7"
url="http://mymachine.domain.com/servlet/com.esri.esrimap.Esrimap"
```

```
service="states" />
        <AVIMSWORKSPACE name="av_ws-6"
url="http://mymachine.domain.com/servlet/com.esri.esrimap.Esrimap"
view="Roads" map="roads21204240" />
        <MOIMSWORKSPACE name="mo_ws-4"
url="http://mymachine.domain.com/scripts/esrimap.dll"
service="GPS_Points" />
        <SHAPEWORKSPACE name="shp_ws-8" directory="<path to CANADA
ESRIDATA" />
      </WORKSPACES>
      <LAYER type="image" name="Background" visible="true" id="0">
        <DATASET name="background" type="image" workspace="mapper_ws-6"
/>
      </LAYER>
      <LAYER type="featureclass" name="States" visible="true" id="1">
        <DATASET name="2" type="polygon" workspace="ifs_ws-7" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL boundarytransparency="1.0"
filltransparency="1.0" fillcolor="255,255,153" boundarycaptype="round"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Provinces" visible="true"
id="2">
        <DATASET name="province" type="polygon" workspace="shp_ws-8" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL boundarytransparency="1.0"
filltransparency="1.0" fillcolor="127,27,27" boundarycaptype="round" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="image" name="Roads" visible="true" id="3">
        <DATASET name="mymachine.domain.com:roads21204240:Roads"
type="image" workspace="av_ws-6" />
        <IMAGEPROPERTIES transparency="0.80" />
      </LAYER>
      <LAYER type="image" name="GPS Points" visible="true" id="4">
        <DATASET name="mymachine.domain.com:GPS_Points" type="image"
workspace="mo_ws-4" />
      </LAYER>
    </MAP>
    <OVERVIEWMAP backgroundcolor="255,255,255"
framefillcolor="255,0,0,80" frameoutlinecolor="255,0,0"
zoomfactor="4.0">
      <LAYERDEF name="States" />
      <LAYERDEF name="Provinces" />
    </OVERVIEWMAP>
    <SCALEBAR backcolor="212,208,200" fontcolor="0,0,0"
mapunits="decimal_degrees" scaleunits="feet" screenunits="inches" />
  </CONFIG>
</ARCXML>
```

The additional elements are:

- IMAGESERVERWORKSPACE. IMAGESERVERWORKSPACE is included in the WORKSPACES section when an Image Service is accessed.

- FEATURESERVERWORKSPACE. FEATURESERVERWORKSPACE is included in the WORKSPACES section when a Feature Service is accessed.
- MOIMSWORKSPACE and AVIMSWORKSPACE. These two elements are included in the WORKSPACES section when a map is accessed from an ArcExplorer-enabled map in MapObjects IMS or from ArcView IMS, respectively.
- SCALEBAR and OVERVIEWMAP. These two elements are client configuration elements that add a scale bar or overview map, respectively.

Each of these elements is described below.

IMAGESERVERWORKSPACE

IMAGESERVERWORKSPACE is used in the WORKSPACES section for Image Service layers.

Using IMAGESERVERWORKSPACE:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>...</ENVIRONMENT>
    <MAP>
      <PROPERTIES>...</PROPERTIES>
      <WORKSPACES>
        <IMAGESERVERWORKSPACE name="mapper_ws-6"
url="http://mymachine.domain.com/servlet/com.esri.esrimap.Esrimap"
service="background" />
      </WORKSPACES>
      <LAYER type="image" name="Background" visible="true" id="0">
        <DATASET name="background" type="image" workspace="mapper_ws-6"
/>
      </LAYER>
    </MAP>
    <OVERVIEWMAP>...</OVERVIEWMAP>
    <SCALEBAR ... />
  </CONFIG>
</ARCXML>
```

References to an Image Service in IMAGESERVERWORKSPACE are made in the *service* attribute. In the above example, the name of the Image Service is "background". The *url* attribute gives the location of the Image Service. In the example, the location is "http://mymachine.domain.com". In addition to the domain, the location of the ArcIMS servlet connector is also included. This information is the same for all ArcIMS sites and is "/servlet/com.esri.esrimap.Esrimap".

For the Image Service layer, references to the ArcIMS service are made in DATASET. The attribute *name* is the same as IMAGESERVERWORKSPACE *service*. In this example, the *name* is "background". The DATASET *type* is always "image". Image

Service layers are always treated as one layer in a viewer configuration file, and the sublayers are not included.

FEATURESERVERWORKSPACE

FEATURESERVERWORKSPACE is used in the WORKSPACES section for Feature Service layers.

Using FEATURESERVERWORKSPACE:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>...</ENVIRONMENT>
    <MAP>
      <PROPERTIES>...</PROPERTIES>
      <WORKSPACES>
        <FEATURESERVERWORKSPACE name="ifs_ws-7"
url="http://mymachine.domain.com/servlet/com.esri.esrimap.Esrimap"
service="states" />
      </WORKSPACES>
      <LAYER type="featureclass" name="States" visible="true" id="1">
        <DATASET name="3" type="polygon" workspace="ifs_ws-7" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL boundarytransparency="1.0"
filltransparency="1.0" fillcolor="255,255,153" boundarycaptype="round"
/>
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
    <OVERVIEWMAP>...</OVERVIEWMAP>
    <SCALEBAR ... />
  </CONFIG>
</ARCXML>
```

References to a Feature Service in FEATURESERVERWORKSPACE are made in the *service* attribute. In the above example, the name of the Feature Service is "States". The *url* attribute is used the same way as with Image Service. In the example, the location of the ArcIMS Servlet Connector is "http://mymachine.domain.com/servlet/com.esri.esrimap.Esrimap".

Each layer in a Feature Service is treated as a separate layer, and each layer can be accessed individually. In the above example, only one layer of a Feature Service has been accessed.

For a Feature Service layer, DATASET is always included, and rendering information is usually included. In DATASET, the attribute *type* is set to the same type as in the Feature Service. In the above example the *type* is "polygon". The attribute *name* is set to the LAYER *id* in the Feature Service. In this example, the LAYER *id* in the Feature Service is "3"; therefore, the DATASET *name* in the viewer configuration file is "3". The following examples highlight the differences between the map configuration file used as input to the Feature Service and the output in the viewer configuration file.

Map configuration file used in the Feature Service:

```
<LAYER type="featureclass" name="States" visible="true" id="3">
  <DATASET name="STATES" type="polygon" workspace="shp_ws-1" />
  <SIMPLERENDERER>
    <SIMPLEPOLYGONSYMBOL boundarytransparency="1.0"
filltransparency="1.0" fillcolor="255,255,153" boundarycaptype="round"
/>
  </SIMPLERENDERER>
</LAYER>
```

Viewer configuration file:

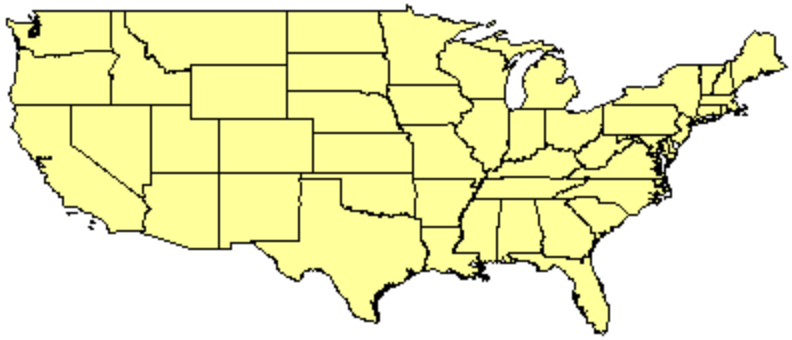
```
<LAYER type="featureclass" name="States" visible="true" id="1">
  <DATASET name="3" type="polygon" workspace="ifs_ws-7" />
  <SIMPLERENDERER>
    <SIMPLEPOLYGONSYMBOL boundarytransparency="1.0"
filltransparency="1.0" fillcolor="255,255,153" boundarycaptype="round"
/>
  </SIMPLERENDERER>
</LAYER>
```

In the map configuration file, the LAYER *id* is "3", which is unique within the map configuration file, and the DATASET *name* is "STATES" and refers to a shapefile in the "shp_ws-1" workspace. In the viewer configuration file, the LAYER *id* is "1", which is unique within the viewer configuration file. The DATASET *name* is "3", which refers to the LAYER *id* in the map configuration file. The *workspace* reference is "ifs_ws-7", which refers to the workspace for the Feature Service. The rest of the layer information in the map and viewer configuration files is identical.

When using ArcExplorer 4 and when permission is given in the ArcIMS Java Viewers, the layer rendering can be changed using the Layer Properties Dialog. In the following example, the States layer is rendered in light yellow by default.

Default viewer configuration file:

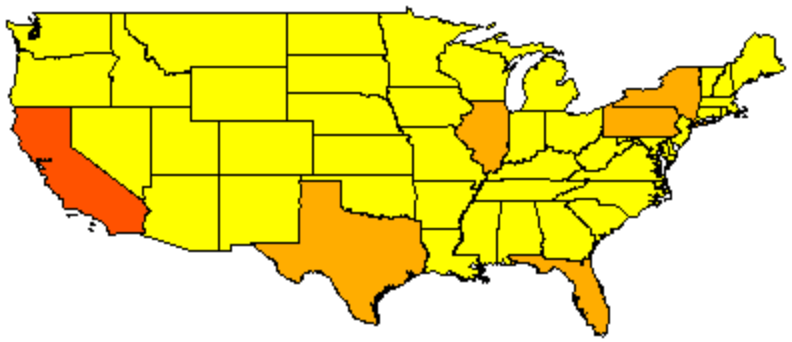
```
<LAYER type="featureclass" name="States" visible="true" id="1">
  <DATASET name="2" type="polygon" workspace="ifs_ws-7" />
  <SIMPLERENDERER>
    <SIMPLEPOLYGONSYMBOL boundarytransparency="1.0"
filltransparency="1.0" fillcolor="255,255,153" boundarycaptype="round"
/>
  </SIMPLERENDERER>
</LAYER>
```



If a user changes the rendering, this new information is stored in the viewer configuration file. The change is local and does not affect the Feature Service. In the next example, the rendering is changed from light yellow to various colors based on the population of each state.

After layer rendering has been changed in the viewer configuration file:

```
<LAYER type="featureclass" name="States" visible="true" id="1">
  <DATASET name="2" type="polygon" workspace="ifs_ws-7" />
  <VALUEMAPRENDERER lookupfield="POP1999">
    <RANGE lower="482025" upper="11351422" label="Less than 11351422">
      <SIMPLEPOLYGONSYMBOL boundarytransparency="1.0"
filltransparency="1.0" fillcolor="255,255,0" boundarycaptype="round" />
    </RANGE>
    <RANGE lower="11351422" upper="22220818" label="11351422 -
22220818">
      <SIMPLEPOLYGONSYMBOL boundarytransparency="1.0"
filltransparency="1.0" fillcolor="255,170,0" boundarycaptype="round" />
    </RANGE>
    <RANGE lower="22220818" upper="33090215" label="22220818 -
33090215">
      <SIMPLEPOLYGONSYMBOL boundarytransparency="1.0"
filltransparency="1.0" fillcolor="255,85,0" boundarycaptype="round" />
    </RANGE>
  </VALUEMAPRENDERER>
</LAYER>
```



Remember that in a viewer configuration file, any rendering changes remain local to the machine that the configuration file resides on. The change does not affect the Feature Service.

AVIMSWORKSPACE

AVIMSWORKSPACE is used in the WORKSPACES section to reference a map generated from ArcView IMS. **Note: AVIMSWORKSPACE has been deprecated in ArcIMS 4.0.**

Using AVIMSWORKSPACE:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>...</ENVIRONMENT>
    <MAP>
      <PROPERTIES>...</PROPERTIES>
      <WORKSPACES>
        <AVIMSWORKSPACE name="av_ws-6"
url="http://mymachine.domain.com/servlet/com.esri.esrimap.Esrimap"
view="Roads" map="roads21204240" />
      </WORKSPACES>
      <LAYER type="image" name="Roads" visible="true" id="3">
        <DATASET name="mymachine.domain.com:roads21204240:Roads"
type="image" workspace="av_ws-6" />
        <IMAGEPROPERTIES transparency="0.80" />
      </LAYER>
    </MAP>
    <OVERVIEWMAP>...</OVERVIEWMAP>
    <SCALEBAR ... />
  </CONFIG>
</ARCXML>
```

AVIMSWOKSPACE references a map in ArcView IMS using the attributes *view* and *map*. The values for these attributes are the ArcView IMS MapName and ViewName, respectively, and can be found in the applet HTML file of an ArcView IMS MapCafé Web site. For example, if the Web site is named "MySite", open "MySiteapplet.html". Look for the parameters MapName and ViewName. For more information on using ArcView IMS with ArcIMS, see *Using ArcIMS*. As with other WORKSPACES types, the *url* attribute references the ArcIMS servlet connector.

The DATASET *name* for the layer references the machine and domain, MapName, and ViewName separated by colons (:). In the above example, the DATASET *name* is "mymachine.domain.com:roads21204240:Roads".

MOIMSWORKSPACE

MOIMSWORKSPACE is used in the WORKSPACES section to reference an ArcExplorer-enabled map generated from MapObjects IMS. **Note: MOIMSWORKSPACE has been deprecated in ArcIMS 4.0.**

Using MOIMSWORKSPACE:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
```

```
  <CONFIG>
    <ENVIRONMENT>...</ENVIRONMENT>
    <MAP>
      <PROPERTIES>...</PROPERTIES>
      <WORKSPACES>
        <MOIMSWORKSPACE name="mo_ws-4"
url="http://mymachine.domain.com/scripts/esrimap.dll"
service="GPS_Points" />
      </WORKSPACES>
      <LAYER type="image" name="GPS Points" visible="true" id="4">
        <DATASET name="mymachine.domain.com:GPS_Points" type="image"
workspace="mo_ws-4" />
      </LAYER>
    </MAP>
    <OVERVIEWMAP>...</OVERVIEWMAP>
    <SCALEBAR ... />
  </CONFIG>
</ARCXML>
```

References to an ArcExplorer-enabled MapObjects IMS map in MOIMSWORKSPACE are made in the *service* attribute. In the above example, the name of the map is "GPS_Points". The *url* attribute gives the location of esrimap.dll located on the machine with MapObjects IMS. ArcExplorer-enabled maps do not work with the ArcIMS Servlet Connector. However, the ArcIMS Servlet Connector can be used with regular MapObjects IMS sites.

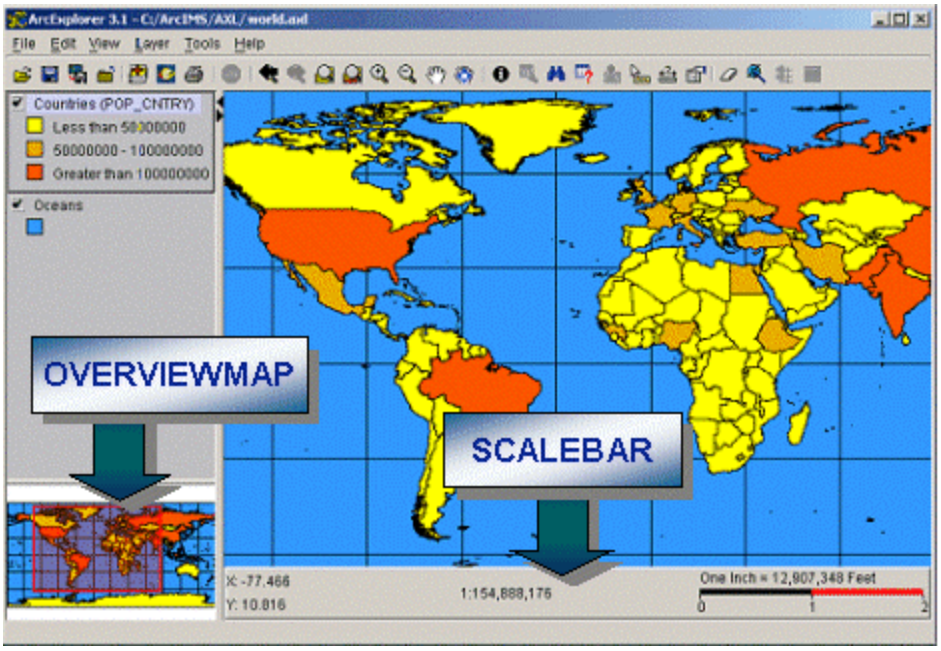
The DATASET *name* references the machine and domain and the MapObjects IMS map name separated by a colon (:). In the above example, the DATASET *name* is "mymachine.domain.com:GPS_Points".

SHAPEWORKSPACE, IMAGEWORKSPACE, and SDEWORKSPACE

SHAPEWORKSPACE, IMAGEWORKSPACE, and SDEWORKSPACE are used when local data is referenced in the viewer configuration file. These WORKSPACES elements reference data exactly the same way as in a map configuration file.

OVERVIEWMAP and SCALEBAR

OVERVIEWMAP and SCALEBAR are client configuration elements and add an overview map or scale bar, respectively. Both elements are saved when using ArcExplorer 4 or an ArcIMS Java Standard Viewer. Only SCALEBAR is saved in an ArcIMS Java Custom Viewer even if an overview map is present. Also, the ArcIMS Java Custom Viewer ignores these elements when a viewer configuration file is read into the viewer.



If an overview map or scale bar are present when a viewer configuration file is saved, the OVERVIEW and SCALEBAR elements are included.

Using OVERVIEWMAP and SCALEBAR:

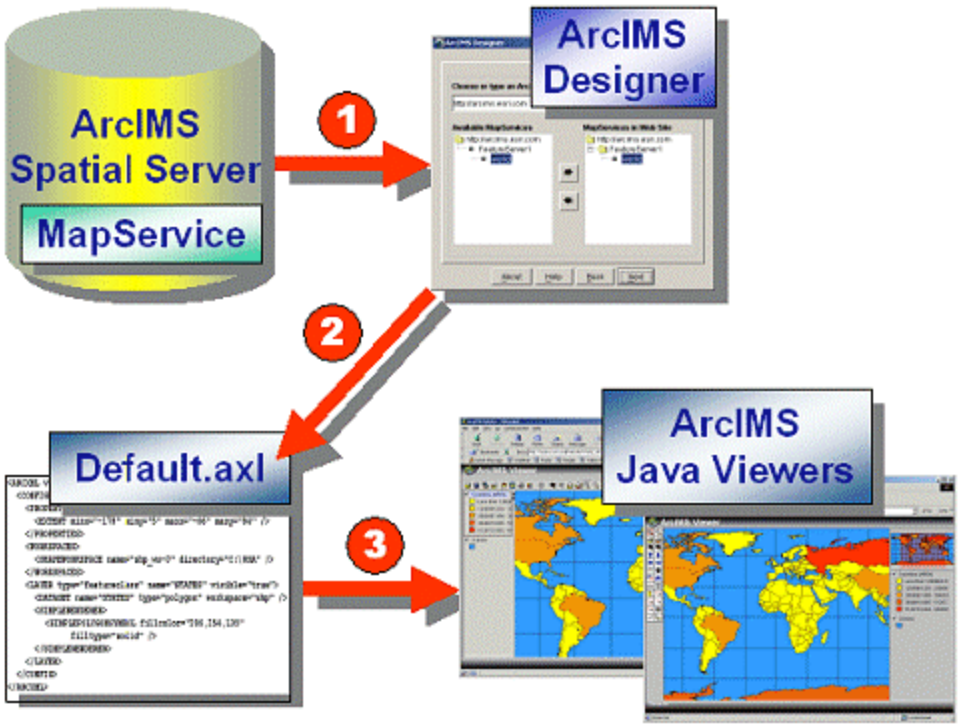
```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>...</ENVIRONMENT>
    <MAP>...</MAP>
    <OVERVIEWMAP backgroundcolor="255,255,255"
framefillcolor="255,0,0,80" frameoutlinecolor="255,0,0"
zoomfactor="4.0">
      <LAYERDEF name="Oceans" />
    <LAYERDEF name="Countries" />
  </OVERVIEWMAP>
    <SCALEBAR backcolor="212,208,200" fontcolor="0,0,0"
mapunits="decimal_degrees" scaleunits="feet" screenunits="inches" />
  </CONFIG>
</ARCXML>
```

If these two elements are included in a map configuration file, they are ignored.

Default.axl: A Special Viewer Configuration File

Default.axl is a special viewer configuration file that is output by ArcIMS Designer when an ArcIMS Java Viewer is created. The diagram below shows the process to create default.axl:

- In Step 1, one or more ArcIMS services are selected during the ArcIMS Designer process.
- In Step 2, when a Web site is generated, one of the output files in the directory is default.axl.
- In Step 3, when an ArcIMS Java Standard or Java Custom Viewer is opened, default.axl is loaded in the viewer. The file contains information on which services should be loaded in the viewer.



Framework of default.axl

Default.axl uses a subset of elements normally found in a viewer configuration file. Its primary purpose is to load ArcIMS service layers specified during the ArcIMS Designer process.

Example default.axl file:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <IMAGESERVERWORKSPACE name="mapper_ws-6"
url="http://mycomputer.domain.com/servlet/com.esri.esrimap.Esrimap"
```

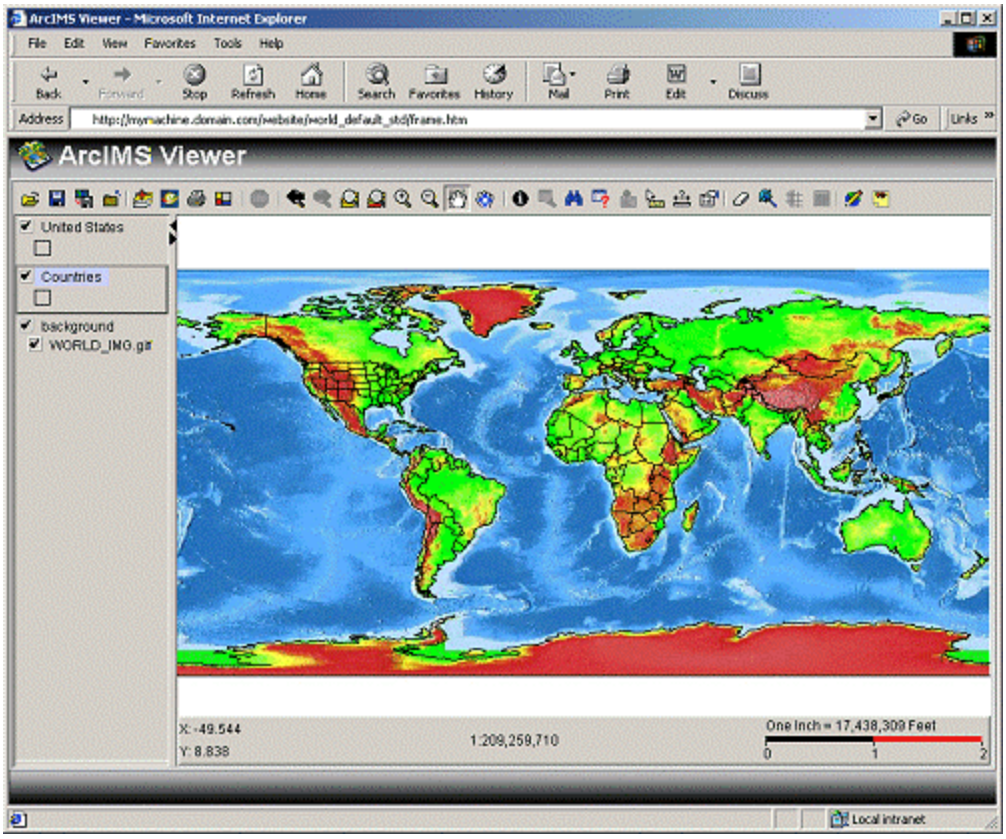
```
service="background" />
        <FEATURESERVERWORKSPACE name="ifs_ws-7"
url="http://mycomputer.domain.com/servlet/com.esri.esrimap.Esrimap"
service="World" />
      </WORKSPACES>
      <LAYER type="image" name="background" visible="true" id="0">
        <DATASET name="background" type="image" workspace="mapper_ws-6"
/>
      </LAYER>
      <LAYER type="featureclass" name="Countries" visible="true"
id="1">
        <DATASET name="0" type="polygon" workspace="ifs_ws-7" />
      </LAYER>
      <LAYER type="featureclass" name="United States" visible="true"
id="2">
        <DATASET name="1" type="polygon" workspace="ifs_ws-7" />
      </LAYER>
    </MAP>
    <SCALEBAR backcolor="212,208,200" fontcolor="0,0,0"
mapunits="decimal_degrees" scaleunits="feet" screenunits="inches" />
  </CONFIG>
</ARCXML>
```

Only elements that are needed to load the ArcIMS services are included. The principal similarities and differences between default.axl and a viewer configuration file are:

- CONFIG and MAP. These elements are always present in a configuration file.
- ENVIRONMENT. The ENVIRONMENT elements are not included in default.axl. Since the information on locale is already associated with the ArcIMS services, the ENVIRONMENT information does not need to be duplicated in default.axl.
- WORKSPACE elements. Only FEATURESERVERWORKSPACE and IMAGESERVERWORKSPACE are added during the ArcIMS Designer process. AVIMSWORKSPACE and MOIMSWORKSPACE can be added manually if a map from ArcView IMS or MapObjects IMS is included in the Web site.
- PROPERTIES. This section is the same as for viewer configuration files. The one exception is the use of ENVELOPE. In the Java Custom Viewer, two ENVELOPE elements may be included. If only one ENVELOPE is included, only the initial extent for the map is included. The following figure shows the initial extent of a world map when the following ENVELOPE is used:

Using ENVELOPE Initial_Extent:

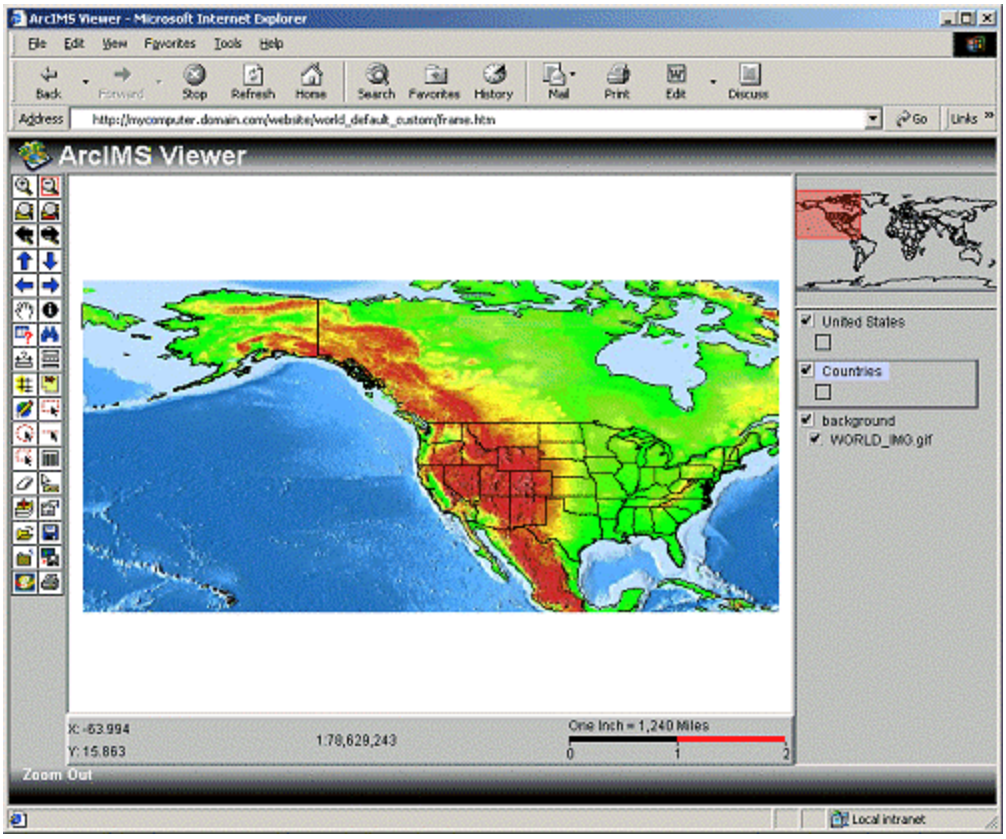
```
<PROPERTIES>
  <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
name="Initial_Extent"/>
  <MAPUNITS units="decimal_degrees" />
<PROPERTIES>
```



If two ENVELOPE elements are included, the first is the map's initial extent, and the second is the maximum zoom limit of the map. These extents are set during the ArcIMS Designer process. The coordinates used in the extent limit represent the maximum limit of data that can be shown. In the next example, the initial extent and the extent limit are set to an envelope that bounds the United States, although a world dataset was used. In the Java Custom Viewer, only the region inside the extent limit displays.

Using ENVELOPE Initial_Extent and Extent_Limit:

```
<PROPERTIES>
  <ENVELOPE minx="-61.1" miny="3.7" maxx="91.7" maxy="61.3"
name="Initial_Extent"/>
  <ENVELOPE minx="-61.1" miny="3.7" maxx="91.7" maxy="61.3"
name="Extent_Limit"/>
  <MAPUNITS units="decimal_degrees" />
<PROPERTIES>
```



- LAYER and DATASET. The rules for assigning attribute values are the same as for viewer configuration files. However, no rendering information is included with the layer.
- SCALEBAR. SCALEBAR is always included in default.axl with Java Standard Viewer Web sites but not with Java Custom Viewer web sites.

Including SCALEBAR:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <MAP>...</MAP>
    <SCALEBAR backcolor="212,208,200" fontcolor="0,0,0"
mapunits="decimal_degrees" scaleunits="feet" screenunits="inches"
/>
  </CONFIG>
</ARCXML>
```

- OVERVIEWMAP. OVERVIEWMAP is not included in default.axl.

Relationship between Feature Services and default.axl

When ArcIMS Designer creates default.axl and a Feature Service is included, one

LAYER is included in default.axl for every LAYER in the Feature Service. Since no rendering is included in default.axl, the rendering defined in the ArcIMS service is what is used by the ArcIMS Java Viewers. Any changes to the rendering in the service are immediately reflected in the ArcIMS Java Viewers.

Caution is needed when adding or deleting layers in Feature Service. If a layer is added to a service but not to default.axl, the ArcIMS Java Viewer will not display the new layer. If a layer is deleted from a service but not default.axl, an error message appears in the ArcIMS Java Viewer because a layer referenced in default.axl cannot be found in the service. In summary, if a layer is added or deleted in a service, it should be added or deleted in default.axl.

Comparing Different CONFIG Files

The table below summarizes the similarities and differences between map configuration files, viewer configuration files, and default.axl.

	Map Config	Viewer Config	Default.axl
Primary purpose	Used as input to ArcIMS services	Saves current map and viewer configuration on the client	Used to load ArcIMS services in an ArcIMS Java Viewer
Created by	ArcIMS Author or XML Editor	ArcExplorer 4 or ArcIMS Java Viewers	ArcIMS Designer
File is located	On server	On client	On server
Data sources	Local network only and must be accessible by ArcIMS Spatial Server	ArcIMS services and local network on client	ArcIMS services only
Valid workspaces	SHAPEWORKSPACE, IMAGEWORKSPACE, SDEWORKSPACE	IMAGESERVERWORKSPACE, FEATURESERVERWORKSPACE, AVIMSWORKSPACE, MOIMSWORKSPACE, SHAPEWORKSPACE, IMAGEWORKSPACE, SDEWORKSPACE	IMAGESERVERWORKSPACE, FEATURESERVERWORKSPACE, AVIMSWORKSPACE, MOIMSWORKSPACE

Using ArcXML Renderers

Introduction

The ArcXML renderers provide the framework for feature symbolization and labeling. You have already been using ArcXML renderers if you have used ArcIMS Author. The output from ArcIMS Author is a map configuration file in ArcXML format. This document assumes you are familiar with ArcIMS Author and the basic structure of a map configuration file. All the examples in this document demonstrate how to use renderers with a map configuration file. The syntax and usage are the same if you want to use the renderers in a request.

Renderers are valid only with Feature and Image Servers. They are not used with the ArcMap Server. However, symbols can be used in requests to the ArcMap Server when they are included in acetate layers.

Each ArcXML renderer includes either a symbol or a label, depending on the renderer. Only one symbol or label is allowed within a renderer. In a generalized form, the relationship between a renderer and a symbol in ArcXML is:

```
<RENDERER>
  <SYMBOL />
</RENDERER>
```

Six renderers are available with ArcIMS:

- SIMPLERENDERER
- SIMPLELABELRENDERER
- VALUEMAPRENDERER
- VALUEMAPLABELRENDERER
- GROUPRENDERER
- SCALEDEPENDENTRENDERER

VALUEMAPRENDERER and VALUEMAPLABELRENDERER also have three child elements used to create value maps:

- RANGE
- EXACT
- OTHER

Only one symbol or label can be used inside a renderer, but the renderers can be used in different combinations to create complex symbols at different scales. The different symbols that can be used inside SIMPLERENDERER and VALUEMAPRENDERER are as follows:

- **Point layers**
  - RASTERMARKERSYMBOL
  - SIMPLEMARKERSYMBOL
  - TRUETYPEMARKERSYMBOL
- **Line layers**
  - All the symbols valid for point layers
  - HASHLINESYMBOL
  - SIMPLELINESYMBOL
- **Polygon layers**
  - All the symbols valid for point and line layers
  - GRADIENTFILLSYMBOL
  - RASTERFILLSYMBOL
  - SIMPLEPOLYGONSYMBOL

Symbols that can be used inside of SIMPLELABELRENDERER and VALUEMAPLABELRENDERER are:

- CALLOUTMARKERSYMBOL
- CHARTSYMBOL
- RASTERSHIELDSYMBOL
- SHIELDSYMBOL
- TEXTSYMBOL

In the examples below, each renderer is briefly defined, and the examples show how the renderer can be used in a map configuration file. The same concepts can be applied to renderers in requests since the syntax is the same.

The examples used in this document build on the same set of data. Two shapefiles are used: CNTRY94 (polygon) and CITIES (point). Each example is an addition or modification to the previous example.

After each renderer has been described, a section is included covering a case study using a set of shapefiles of New York City. This study is a good example of a map configuration file using all the renderers.

SIMPLERENDERER

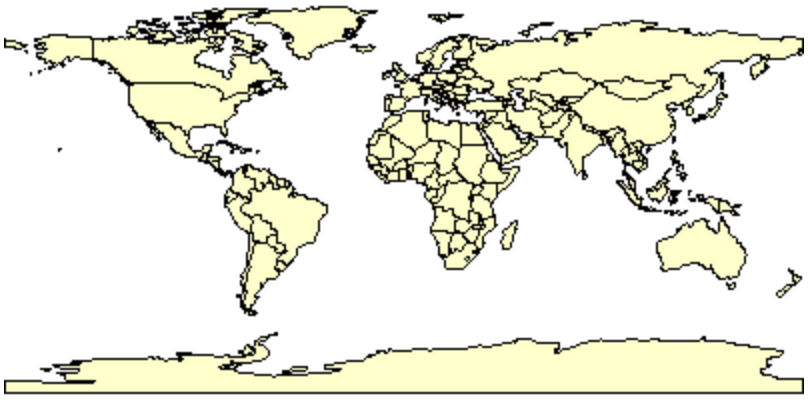
SIMPLERENDERER is used to provide the framework for filling polygons, drawing lines, and displaying points in a layer using one symbol.

In the following example, a SIMPLERENDERER is used to describe how polygons in CNTRY94 are drawn. The symbol instructions are inside SIMPLEPOLYGONSYMBOL. In this case, a SIMPLEPOLYGONSYMBOL is used with a solid *filltype* and a light yellow *fillcolor*.

Using SIMPLERENDERER:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="C:\ESRIDATA\WORLD"
/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="1">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="255,255,204" filltype="solid"
/>
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

The resulting map shows all countries rendered in light yellow.



GROUPRENDERER

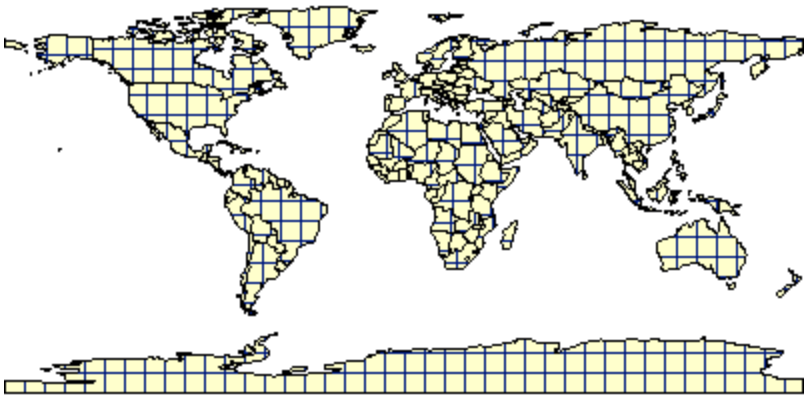
GROUPRENDERER is required when two or more renderers are used to describe a layer. For example, if you want to create a multipart symbol, you can use GROUPRENDERER to group together two SIMPLERENDERERS. Another common use for GROUPRENDERER is when both symbols and labels are applied to a layer (discussed in the next section).

In the next example, a GROUPRENDERER is used to create a multipart symbol. Inside the GROUPRENDERER are two SIMPLERENDERERS using SIMPLEPOLYGONSYMBOL. The underlying *filltype* is solid with a cross *filltype* on top. Note that the renderers and symbols are processed in the order they are read. Since the solid SIMPLEPOLYGONSYMBOL is first, it is the bottom layer.

Using SIMPLERENDERERs inside a GROUPRENDERER:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="C:\ESRIDATA\WORLD"
/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="1">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL fillcolor="255,255,204"
filltype="solid" />
          </SIMPLERENDERER>
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL fillcolor="0,27,127" filltype="cross"
/>
          </SIMPLERENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

The resulting map shows all countries rendered first in light yellow with a cross pattern on top.



SIMPLELABELRENDERER

SIMPLELABELRENDERER is used to label the features in a layer using data from a specified field in the database. In this next example, SIMPLELABELRENDERER is added to the CNTRY94 layer. The field used for labeling is NAME.

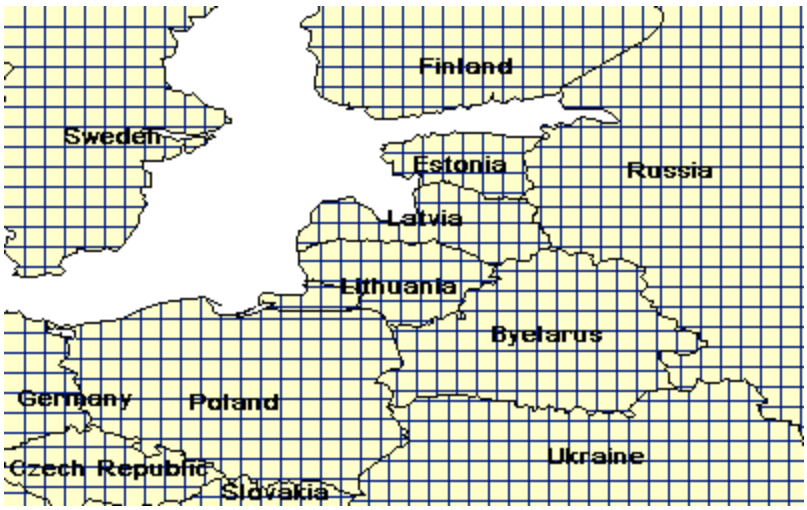
In order to both label and symbolize a layer, you must use a GROUPRENDERER. The two SIMPLERENDERER elements from the previous example are grouped together with SIMPLELABELRENDERER using a GROUPRENDERER.

Using SIMPLERENDERER and SIMPLELABELRENDERER:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="C:\ESRIDATA\WORLD"
/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="1">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL fillcolor="255,255,204"
```

```
filltype="solid" />
          </SIMPLERENDERER>
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL fillcolor="0,27,127" filltype="cross"
/>
          </SIMPLERENDERER>
          <SIMPLELABELRENDERER field="NAME">
            <TEXTSYMBOL fontstyle="bold" fontsize="12" />
          </SIMPLELABELRENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

In the returned map, labels are included as part of the rendering for each country.



If you want a layer to be labeled without any symbology, you can do this by including only a SIMPLELABELRENDERER for the layer. In this case you do not need SIMPLERENDERER or GROUPRENDERER. The following example uses only SIMPLELABELRENDERER for the CNTRY94 layer.

Using only SIMPLELABELRENDERER:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
```

```
        <SHAPEWORKSPACE name="shp_ws-0" directory="C:\ESRIDATA\WORLD"
/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="1">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <SIMPLELABELRENDERER field="NAME">
          <TEXTSYMBOL fontstyle="bold" fontsize="12" />
        </SIMPLELABELRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

The returned map includes only labels. In the following figure, the country boundaries are included for reference only.



SCALEDEPENDENTRENDERER

Setting scales for displaying data is a very important part of designing a map configuration file. There are two ways to use scale to control the display of features. The first way is to set a scale factor to turn a layer on and off depending on the scale. The scale factor is set in LAYER using *minscale* and *maxscale* attributes. Setting this factor is particularly important for layers with detailed data. In the following example, a Cities layer has been added to the map configuration file. This layer does not display until the scale is less than 1:50000000.

The second way to use scale is to use a SCALEDEPENDENTRENDERER to change the symbology of a layer when a specified scale is met. The layer always displays, but as you zoom in the symbology changes. For example, in the first SCALEDEPENDENTRENDERER below, in the Cities layer, when the scale is between 1:25000000 and 1:50000000, the location of the city is marked with a circle using SIMPLEMARKERSYMBOL inside a SIMPLERENDERER. In the second SCALEDEPENDENTRENDERER, when the scale threshold is less than 1:25000000, the markers change to a larger star.

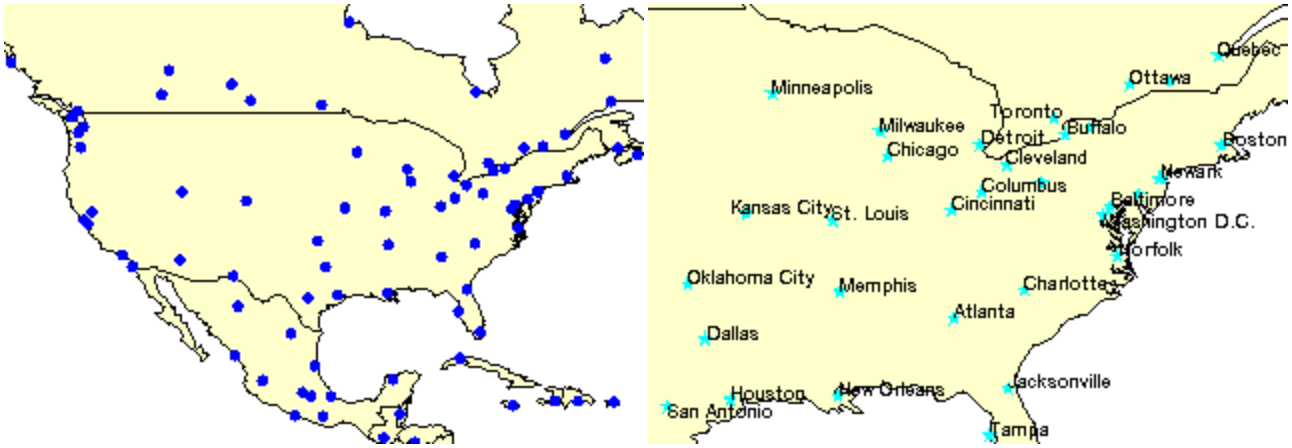
A third SCALEDEPENDENTRENDERER is used to turn on labeling for Cities when the scale is 1:25000000. TEXTSYMBOL is used inside a SIMPLELABELRENDERER. The three SCALEDEPENDENTRENDERERS must be grouped inside a GROUPRENDERER.

Using SCALEDEPENDENTRENDERER:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
        <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="C:\ESRIDATA\WORLD"
/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="1">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="255,255,204" filltype="solid"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Cities" visible="true" id="2"
maxscale="1:50000000">
        <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <SCALEDEPENDENTRENDERER lower="1:25000000"
upper="1:50000000">
            <SIMPLERENDERER>
              <SIMPLEMARKERSYMBOL color="0,0,255" width="8"
type="circle" />
            </SIMPLERENDERER>
          </SCALEDEPENDENTRENDERER>
          <SCALEDEPENDENTRENDERER upper="1:25000000">
            <SIMPLERENDERER>
              <SIMPLEMARKERSYMBOL color="0,255,255" width="16"
type="star" />
            </SIMPLERENDERER>
          </SCALEDEPENDENTRENDERER>
          <SCALEDEPENDENTRENDERER upper="1:25000000">
            <SIMPLELABELRENDERER field="NAME">
              <TEXTSYMBOL fontstyle="regular" fontsize="10" />
            </SIMPLELABELRENDERER>
          </SCALEDEPENDENTRENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
```

```
</ARCXML>
```

The following figures show a map at two different scales. The map on the left is 1:50000000 and shows each city with a circle. The map on the right is at 1:25000000 and shows each city with a star and label.



VALUEMAPRENDERER

VALUEMAPRENDERER is used to classify features in a layer according to values in a specified field in the database. Based on this field, features in different classifications can be rendered differently. Value maps can be categorized using:

- RANGE for graduated symbols maps. With RANGE, a range of values is used to define the category.
- EXACT for unique symbols maps. With EXACT, the values in the specified field must match exactly.
- OTHER, for a catch-all category for features that do not fit in any other RANGE or EXACT category. OTHER is optional to use in a value map and, if it is not used, symbols that do not fall into one of the categories are not drawn.

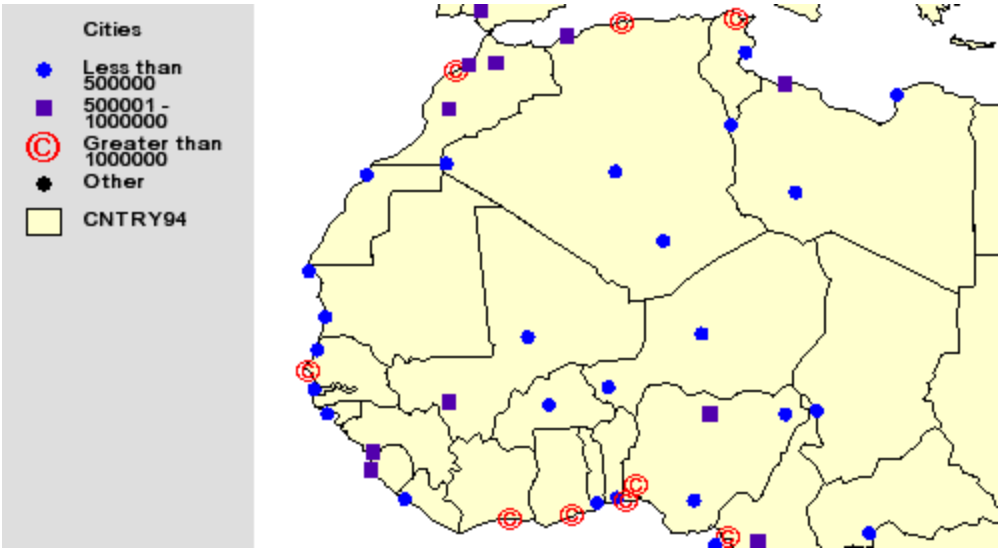
In the following example, the Cities layer (which does not display until the scale is less than 1:50000000) is broken into three RANGE categories using VALUEMAPRENDERER with *lookupfield* set to POPULATION. When assigning symbols, it is permissible to use different symbol elements within the same value map. The following example uses both SIMPLEMARKERSYMBOL and TRUETYPEMARKERSYMBOL.

The first RANGE category is for cities with fewer than 500,000 in population and uses a circle for the SIMPLEMARKERSYMBOL. The second RANGE category is for a population between 500,000 and 1,000,000 and uses a square marker. The third RANGE category is for a population greater than 1,000,000, and the marker for this category is a TRUETYPEMARKERSYMBOL.

Using VALUEMAPRENDERER:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-6" directory="C:\ESRIDATA\WORLD"
/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="0">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-6" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="255,255,204" filltype="solid"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Cities" visible="true" id="1"
maxscale="1:50000000">
        <DATASET name="CITIES" type="point" workspace="shp_ws-6" />
        <VALUEMAPRENDERER lookupfield="POPULATION">
          <RANGE lower="0" upper="500000" label="Less than 500000">
            <SIMPLEMARKERSYMBOL color="0,0,255" width="8" type="circle"
/>
          </RANGE>
          <RANGE lower="500001" upper="1000000" label="500001 -
1000000">
            <SIMPLEMARKERSYMBOL color="85,0,170" width="8"
type="square" />
          </RANGE>
          <RANGE lower="1000001" upper="23620001" label="Greater than
1000000">
            <TRUETYPEMARKERSYMBOL fontstyle="regular" character="169"
font="Arial" fontcolor="255,0,0" fontsize="20"  />
          </RANGE>
          <OTHER>
            <SIMPLEMARKERSYMBOL width="8" />
          </OTHER>
        </VALUEMAPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

The resulting map shows the three categories of cities based on population.



Creating multipart symbols

Multipart symbols can be created for one or more categories of a value map. To do this, additional VALUEMAPRENDERERs are used with only the RANGE or EXACT categories you are interested in. In the next example, the marker for cities with a population less than 500,000 is a three-part symbol:

- The base part is a large blue circle.
- The middle part is a yellow star.
- The top part is a small red circle.

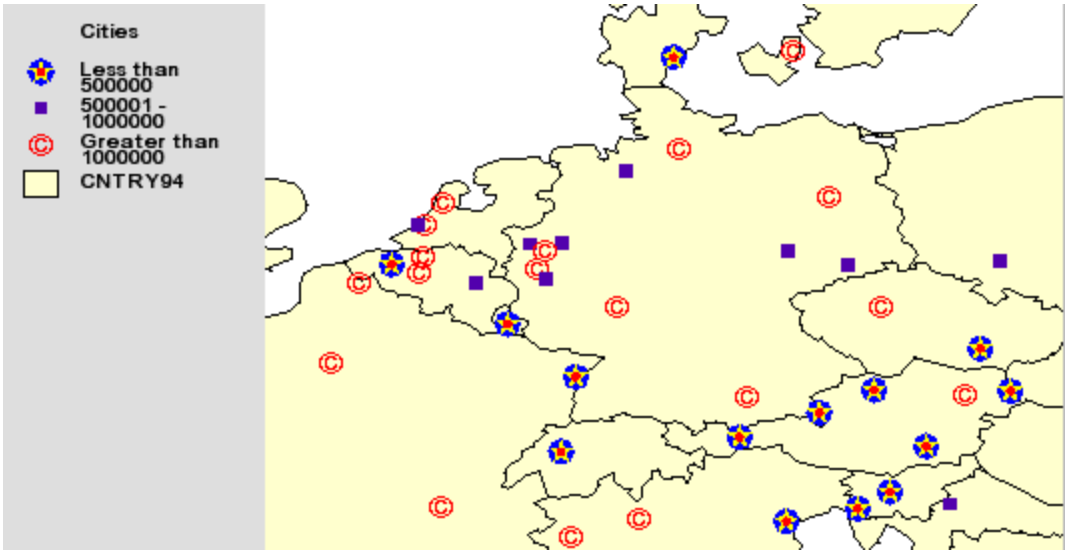
To make this symbol, three VALUEMAPRENDERERs are used. In the first one, the blue circle is rendered. The other RANGE categories are also included for cities with a population between 500,000 and 1,000,000 and with a population greater than 1,000,000. In the second and third VALUEMAPRENDERERs, the only RANGE included is for a population less than 500,000. All three VALUEMAPRENDERERs are grouped inside a GROUPRENDERER.

Using VALUEMAPRENDERER to create a multipart symbol:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-6" directory="C:\ESRIDATA\WORLD"
```

```
/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="0">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-6" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="255,255,204" filltype="solid"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Cities" visible="true" id="1"
maxscale="1:50000000">
        <DATASET name="CITIES" type="point" workspace="shp_ws-6" />
        <GROUPRENDERER>
          <VALUEMAPRENDERER lookupfield="POPULATION">
            <RANGE lower="0" upper="500000" label="Less than 500000">
              <!--blue circle-->
              <SIMPLEMARKERSYMBOL color="0,0,255" width="16"
type="circle" />
            </RANGE>
            <RANGE lower="500001" upper="1000000" label="500001 -
1000000">
              <SIMPLEMARKERSYMBOL color="85,0,170" width="7"
type="square" />
            </RANGE>
            <RANGE lower="1000001" upper="23620001" label="Greater than
1000000">
              <TRUETYPEMARKERSYMBOL fontstyle="regular" character="169"
font="Arial" fontcolor="255,0,0" fontsize="20"  />
            </RANGE>
          </VALUEMAPRENDERER>

          <VALUEMAPRENDERER lookupfield="POPULATION">
            <RANGE lower="0" upper="500000" label="Less than 500000">
              <!--yellow star-->
              <SIMPLEMARKERSYMBOL color="255,255,0" width="16"
type="star" />
            </RANGE>
          </VALUEMAPRENDERER>

          <VALUEMAPRENDERER lookupfield="POPULATION">
            <RANGE lower="0" upper="500000" label="Less than 500000">
              <!--red circle-->
              <SIMPLEMARKERSYMBOL color="255,0,0" width="4"
type="circle" />
            </RANGE>
          </VALUEMAPRENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

The resulting map shows cities with a multipart symbol for cities with a population less than 500,000 plus the two additional categories of cities between 500,000 and 1,000,000 and cities greater than 1,000,000.



VALUEMAPRENDERER inside a SCALEDEPENDENTRENDERER

A VALUEMAPRENDERER can be used inside of a SCALEDEPENDENTRENDERER. In the following example, the Cities layer is divided into two SCALEDEPENDENTRENDERERs. When the scale is between 1:25000000 and 1:50000000, all cities are rendered with a blue circle using a SIMPLERENDERER. When the scale is less than 1:25000000, the value map from the previous example is used for rendering each city. City symbols are categorized by population. GROUPRENDERER is used around both SCALEDEPENDENTRENDERER elements and to group the VALUEMAPRENDERER elements.

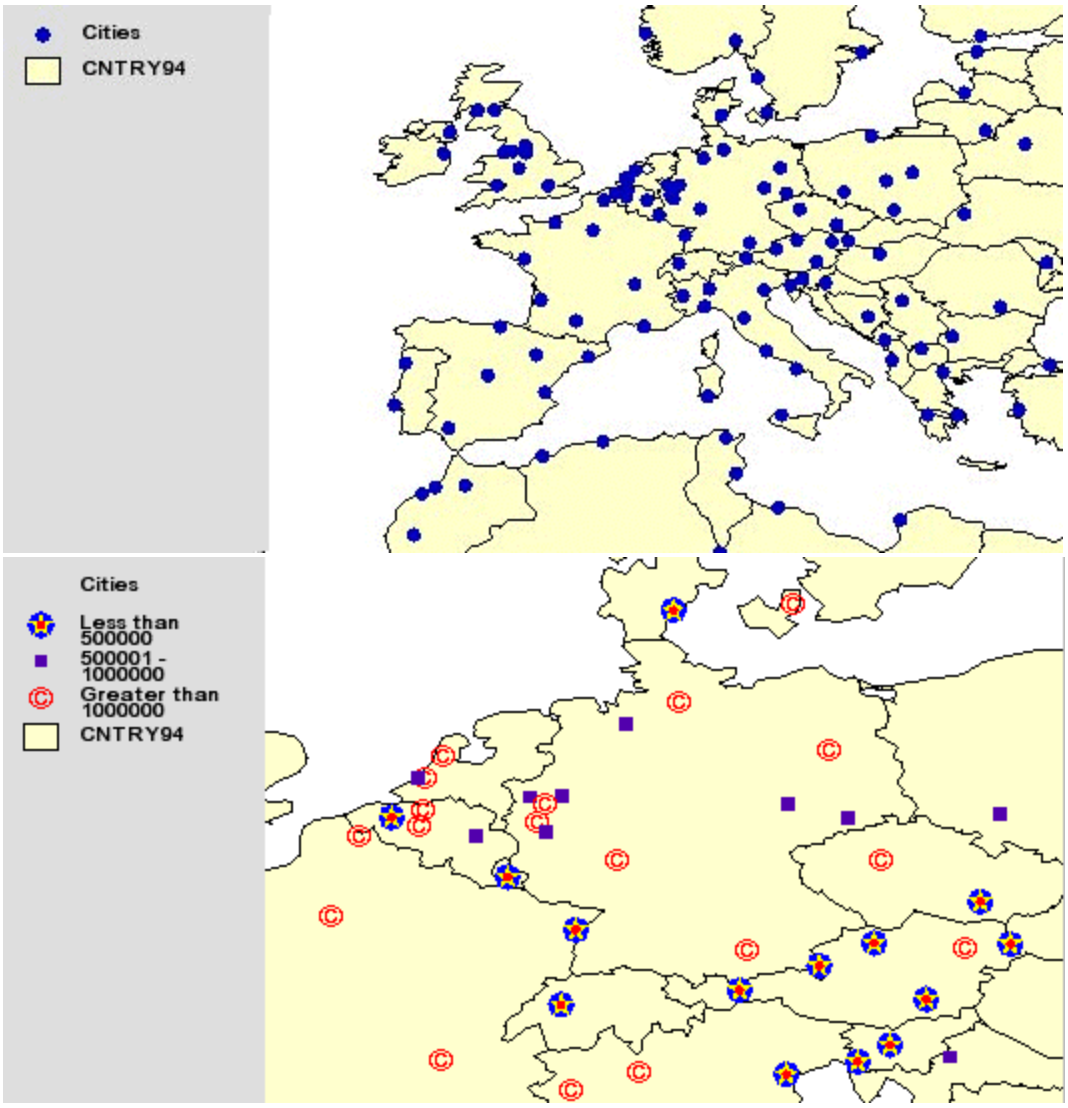
Using VALUEMAPRENDERER inside a SCALEDEPENDENTRENDERER:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-6" directory="C:\ESRIDATA\WORLD"
/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="0">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-6" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="255,255,204" filltype="solid"
/>
        </SIMPLERENDERER>
```

```
      </LAYER>
      <LAYER type="featureclass" name="Cities" visible="true" id="1"
maxscale="1:50000000">
        <DATASET name="CITIES" type="point" workspace="shp_ws-6" />
        <GROUPRENDERER>
          <SCALEDEPENDENTRENDERER lower="1:25000000" >
            <SIMPLERENDERER>
              <SIMPLEMARKERSYMBOL color="0,0,190" width="8"
type="circle" />
            </SIMPLERENDERER>
          </SCALEDEPENDENTRENDERER>

          <SCALEDEPENDENTRENDERER upper="1:25000000" >
            <GROUPRENDERER>
              <VALUEMAPRENDERER lookupfield="POPULATION">
                <RANGE lower="0" upper="500000" label="Less than
500000">
                  <SIMPLEMARKERSYMBOL color="0,0,255" width="16"
type="circle" />
                </RANGE>
                <RANGE lower="500001" upper="1000000" label="500001 -
1000000">
                  <SIMPLEMARKERSYMBOL color="85,0,170" width="7"
type="square" />
                </RANGE>
                <RANGE lower="1000001" upper="23620001" label="Greater
than 1000000">
                  <TRUETYPEMARKERSYMBOL fontstyle="regular"
character="169" font="Arial" fontcolor="255,0,0" fontsize="20" />
                </RANGE>
              </VALUEMAPRENDERER>

              <VALUEMAPRENDERER lookupfield="POPULATION">
                <RANGE lower="0" upper="500000" label="Less than
500000">
                  <SIMPLEMARKERSYMBOL color="255,255,0" width="16"
type="star" />
                </RANGE>
              </VALUEMAPRENDERER>

              <VALUEMAPRENDERER lookupfield="POPULATION">
                <RANGE lower="0" upper="500000" label="Less than
500000">
                  <SIMPLEMARKERSYMBOL color="255,0,0" width="4"
type="circle" />
                </RANGE>
              </VALUEMAPRENDERER>
            </GROUPRENDERER>
          </SCALEDEPENDENTRENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

The following two figures show the same map at two scales. The first map is at

1:30000000 and shows all cities with a small circle. The second map is at 1:12000000 and symbolizes cities based on population.



VALUEMAPLABELRENDERER

VALUEMAPLABELRENDERER applies label formats based on different classifications in a field in a database. Labels can be categorized using RANGE, EXACT, or OTHER, similar to VALUEMAPRENDERER for feature symbolization.

In the following example, the Cities layer (which does not display until the scale is less than 1:50000000) is labeled based on three RANGEs. The RANGEs are cities with a population less than 500,000, a population of 500,000 to 1,000,000, and a population greater than 1,000,000. For each category, TEXTSYMBOL is used, but instructions for labeling are different. Small cities use the Tahoma font. Medium-sized cities use an italic Arial font with glow. Large cities use a bold italic Times New Roman font with glow and shadow.

All cities are rendered with a circle using SIMPLEMARKERSYMBOL inside a SIMPLERENDERER. Both the SIMPLERENDERER and VALUEMAPLABELRENDERER are grouped inside a GROUPRENDERER.

Using VALUEMAPLABELRENDERER:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-6" directory="C:\ESRIDATA\WORLD"
/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="0">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-6" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="255,255,204" filltype="solid"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Cities" visible="true" id="1"
maxscale="1:50000000">
        <DATASET name="CITIES" type="point" workspace="shp_ws-6" />
        <GROUPRENDERER>
          <SIMPLERENDERER>
            <SIMPLEMARKERSYMBOL color="0,0,190" width="8" type="circle"
/>
          </SIMPLERENDERER>

          <VALUEMAPLABELRENDERER lookupfield="POPULATION"
labelfield="NAME" >
            <RANGE lower="0" upper="500000" >
              <TEXTSYMBOL font="Tahoma" fontstyle="regular"
fontsize="10" />
            </RANGE>
            <RANGE lower="500001" upper="1000000" >
              <TEXTSYMBOL font="Arial" fontstyle="italic" fontsize="12"
glowing="125,125,125" />
            </RANGE>
            <RANGE lower="1000001" upper="23620001" >
              <TEXTSYMBOL font="Times New Roman" fontstyle="bolditalic"
fontsize="14" glowing="255,255,0" shadow="0,0,0" />
            </RANGE>
          </VALUEMAPLABELRENDERER>
        </GROUPRENDERER>
      </LAYER>
```

```
    </MAP>
  </CONFIG>
</ARCXML>
```

In the resulting map, all cities are identified by a blue marker. Based on the population of the city, one of three label formats is used.



Using VALUEMAPRENDERER and VALUEMAPLABELRENDERER together

VALUEMAPRENDERER and VALUEMAPLABELRENDERER can be used within the same GROUPRENDERER. In the next example, VALUEMAPRENDERER replaces the SIMPLERENDERER information for the Cities layer. The following symbology and labels are used based on city population:

City Population Range	Symbol	Label
Less than 500,000	Circle	Tahoma font
500,000 to 1,000,000	Square	Italic Arial font with glow
Greater than 1,000,000	TrueType symbol	Bold italic Times New Roman font with glow and shadow

Using VALUEMAPRENDERER and VALUEMAPLABELRENDERER together:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
```

```
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-6" directory="C:\ESRIDATA\WORLD"
/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="0">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-6" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltransparency="1.0"
fillcolor="255,255,204" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Cities" visible="true" id="1"
maxscale="1:50000000">
        <DATASET name="CITIES" type="point" workspace="shp_ws-6" />
        <GROUPRENDERER>
          <VALUEMAPRENDERER lookupfield="POPULATION">
            <RANGE lower="0" upper="500000" label="Less than 500000">
              <SIMPLEMARKERSYMBOL color="0,0,255" width="5"
type="circle" />
            </RANGE>
            <RANGE lower="500001" upper="1000000" label="500001 -
1000000">
              <SIMPLEMARKERSYMBOL color="85,0,170" width="7"
type="square" />
            </RANGE>
            <RANGE lower="1000001" upper="23620001" label="Greater than
1000000">
              <TRUETYPEMARKERSYMBOL fontstyle="regular" character="169"
font="Arial" fontcolor="255,0,0" fontsize="20"  />
            </RANGE>
          </VALUEMAPRENDERER>

          <VALUEMAPLABELRENDERER lookupfield="POPULATION"
labelfield="NAME" >
            <RANGE lower="0" upper="500000" >
              <TEXTSYMBOL font="Tahoma" fontstyle="regular"
fontsize="10" />
            </RANGE>
            <RANGE lower="500001" upper="1000000" >
              <TEXTSYMBOL font="Arial" fontstyle="italic" fontsize="12"
glowing="125,125,125" />
            </RANGE>
            <RANGE lower="1000001" upper="23620001" >
              <TEXTSYMBOL font="Times New Roman" fontstyle="bolditalic"
fontsize="14" glowing="255,255,0" shadow="0,0,0" />
            </RANGE>
          </VALUEMAPLABELRENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

In the resulting map, the marker and label assigned to each city are based on the population.



Limiting labels to a subgroup of features

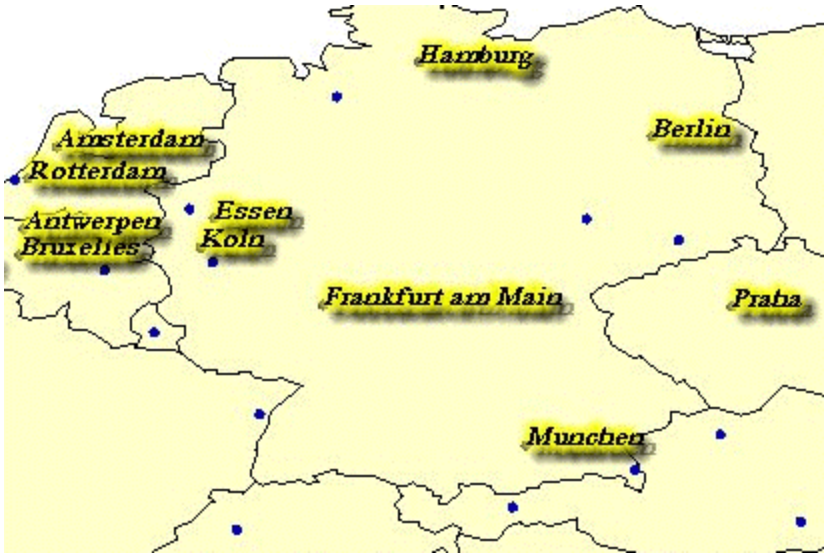
VALUEMAPLABELRENDERER can also be used to limit labeling to a subgroup of features. In the next example, only cities with a population greater than 1000000 are labeled. To do this, only one RANGE category is used inside the VALUEMAPLABELRENDERER. If a city's population does not fall within this range, then it is not labeled.

Using VALUEMAPLABELRENDERER to label a subgroup of features:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-6" directory="C:\ESRIDATA\WORLD"
/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="0">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-6" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="255,255,204" filltype="solid"
/>
        </SIMPLERENDERER>
```

```
      </LAYER>
      <LAYER type="featureclass" name="Cities" visible="true" id="1"
maxscale="1:50000000">
        <DATASET name="CITIES" type="point" workspace="shp_ws-6" />
        <GROUPRENDERER>
          <SIMPLERENDERER>
            <SIMPLEMARKERSYMBOL color="0,0,190" width="6" type="circle"
/>
          </SIMPLERENDERER>
          <VALUEMAPLABELRENDERER lookupfield="POPULATION"
labelfield="NAME" >
            <RANGE lower="1000000" upper="23620001" >
              <TEXTSYMBOL font="Times New Roman" fontstyle="bolditalic"
fontsize="14" glowing="255,255,0" shadow="0,0,0" />
            </RANGE>
          </VALUEMAPLABELRENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

In the resulting map, only cities with a population greater than 1000000 are labeled.



VALUEMAPLABELRENDERER inside a SCALEDEPENDENTRENDERER

VALUEMAPLABELRENDERER can be used inside a SCALEDEPENDENTRENDERER. In the next example, only cities with a population greater than 1000000 are labeled when the scale is between 1:25000000 and 1:50000000. When the scale is less than 1:25000000, all cities are labeled according to their population. A GROUPRENDERER is used around the two SCALEDEPENDENTRENDERER elements.

The following table summarizes the labels used at different scales:

City Population Range	Label
When scale is greater than 1:25000000:	
Greater than 1,000,000	Bold italic Times New Roman font with Glow and Shadow
When scale is less than 1:25000000:	
Less than 500,000	Tahoma font
500,000 to 1,000,000	Italic Arial font with glow
Greater than 1,000,000	Bold italic Times New Roman font with Glow and Shadow

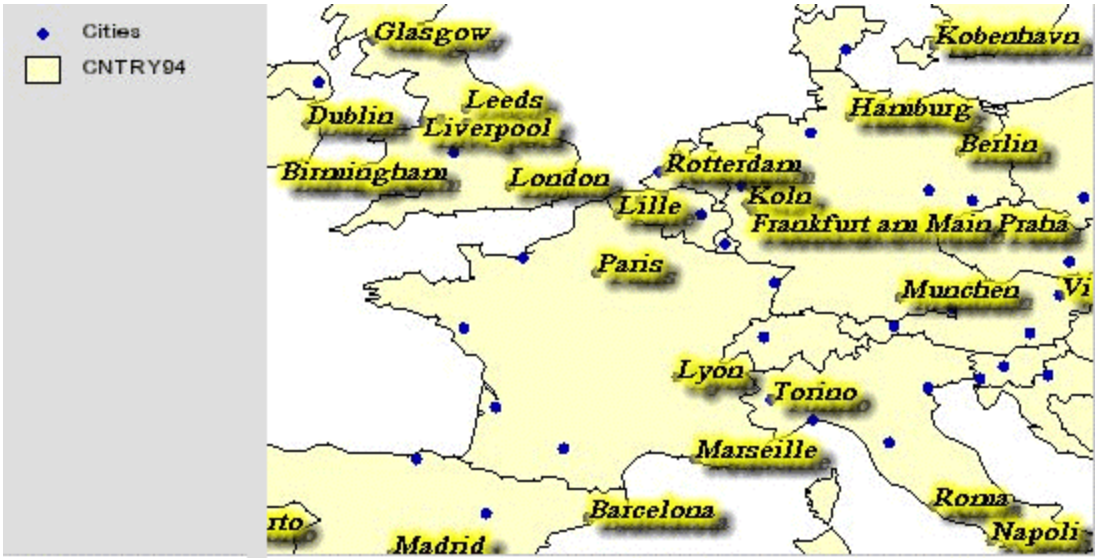
Using VALUEMAPLABELRENDERER inside a SCALEDEPENDENTRENDERER:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-6" directory="C:\ESRIDATA\WORLD"
/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="0">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-6" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="255,255,204" filltype="solid"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Cities" visible="true" id="1"
maxscale="1:50000000">
        <DATASET name="CITIES" type="point" workspace="shp_ws-6" />
        <GROUPRENDERER> <!--Around all renderers-->
          <SCALEDEPENDENTRENDERER lower="1:25000000" > <!--labeling and
rendering when the scale is greater than 1:25000000-->
            <GROUPRENDERER>
              <SIMPLERENDERER>
                <SIMPLEMARKERSYMBOL color="0,0,190" width="6"
type="circle" />
              </SIMPLERENDERER>

              <!--Only label cities with population greater than
1,000,000-->
              <VALUEMAPLABELRENDERER lookupfield="POPULATION"
labelfield="NAME" >
                <RANGE lower="1000001" upper="23620001" >
                  <TEXTSYMBOL font="Times New Roman" fontstyle="bold"
fontsize="14" glowing="255,255,0" shadow="0,0,0" />
                </RANGE>
```

```
            </VALUEMAPLABELRENDERER>
          </GROUPRENDERER>
        </SCALEDEPENDENTRENDERER>

        <SCALEDEPENDENTRENDERER upper="1:25000000" ><!--Labeling and
rendering when the scale is less than 1:25000000-->
          <GROUPRENDERER>
            <!--Render cities based on population-->
            <VALUEMAPRENDERER lookupfield="POPULATION">
              <RANGE lower="0" upper="500000" label="Less than
500000">
                <SIMPLEMARKERSYMBOL color="0,0,255" width="16"
type="circle" />
              </RANGE>
              <RANGE lower="500001" upper="1000000" label="500001 -
1000000">
                <SIMPLEMARKERSYMBOL color="85,0,170" width="7"
type="square" />
              </RANGE>
              <RANGE lower="1000001" upper="23620001" label="Greater
than 1000000">
                 <TRUETYPEMARKERSYMBOL fontstyle="regular"
character="169" font="Arial" fontcolor="255,0,0" fontsize="20"  />
              </RANGE>
            </VALUEMAPRENDERER>

            <VALUEMAPRENDERER lookupfield="POPULATION">
              <RANGE lower="0" upper="500000" label="Less than
500000">
                <SIMPLEMARKERSYMBOL color="255,255,0" width="16"
type="star" />
              </RANGE>
            </VALUEMAPRENDERER>

            <VALUEMAPRENDERER lookupfield="POPULATION">
              <RANGE lower="0" upper="500000" label="Less than
500000">
                <SIMPLEMARKERSYMBOL color="255,0,0" width="4"
type="circle" />
              </RANGE>
            </VALUEMAPRENDERER>

            <!--Label all cities based on population-->
            <VALUEMAPLABELRENDERER lookupfield="POPULATION"
labelfield="NAME" >
              <RANGE lower="0" upper="500000" >
                <TEXTSYMBOL font="Tahoma" fontstyle="regular"
fontsize="10" />
              </RANGE>
              <RANGE lower="500001" upper="1000000" >
                <TEXTSYMBOL font="Arial" fontstyle="italic"
fontsize="12" glowing="125,125,125" />
              </RANGE>
              <RANGE lower="1000001" upper="23620001" >
                <TEXTSYMBOL font="Times New Roman"
fontstyle="bolditalic" fontsize="14" glowing="255,255,0" shadow="0,0,0"
/>
              </RANGE>
```

```
            </VALUEMAPLABELRENDERER>
          </GROUPRENDERER>
        </SCALEDEPENDENTRENDERER>
      </GROUPRENDERER>
    </LAYER>
  </MAP>
 </CONFIG>
</ARCXML>
```

The following two figures are the same map at two scales. The scale for the first map is 1:40000000. At this scale, only cities with a population greater than 1,000,000 are labeled. The scale for the second map is 1:8000000. All cities are labeled based on population.





A Case Study

The following case study uses New York City street data to apply a combination of renderers for displaying these roads with different rendering at different scales. The map configuration file is shown here in its entirety. Take a look at this file to get a general overview. Each of the layers is discussed below.

Entire map configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-74.066" miny="40.689" maxx="-73.823"
maxy="40.883" name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>

      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="c:\nycity\data" />
      </WORKSPACES>

      <LAYER type="featureclass" name="Background" visible="true"
id="0" >
        <DATASET name="nyc_back" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="145,239,254" filltype="solid"
/>
        </SIMPLERENDERER>
      </LAYER>

      <LAYER type="featureclass" name="City Limits" visible="true"
id="1" >
        <DATASET name="nyc_bound" type="polygon" workspace="shp_ws-0"
/>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="255,249,165" filltype="solid"
/>
        </SIMPLERENDERER>
      </LAYER>

      <LAYER type="featureclass" name="Parks" visible="true" id="2" >
        <DATASET name="nyc_parks" type="polygon" workspace="shp_ws-0"
/>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="135,184,114" filltype="solid"
/>
        </SIMPLERENDERER>
      </LAYER>

      <LAYER type="featureclass" name="Water" visible="true" id="3" >
        <DATASET name="nyc_water" type="polygon" workspace="shp_ws-0"
/>
        <SIMPLERENDERER>
```

```
          <SIMPLEPOLYGONSYMBOL fillcolor="145,239,254" filltype="solid"
/>
        </SIMPLERENDERER>
      </LAYER>

      <LAYER type="featureclass" name="Roads" visible="true" id="4"
maxscale="1:35000">
        <DATASET name="nyc_roads" type="line" workspace="shp_ws-0" />

        <GROUPRENDERER>
          <SCALEDEPENDENTRENDERER lower="1:21300" >

            <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
              <EXACT value="Freeway">
                <SIMPLELINESYMBOL type="solid" width="3"
color="255,0,0" />
              </EXACT>
              <EXACT value="Street">
                <SIMPLELINESYMBOL type="solid" width="1"
color="0,0,255" />
              </EXACT>
            </VALUEMAPRENDERER>

          </SCALEDEPENDENTRENDERER>

          <SCALEDEPENDENTRENDERER upper="1:21300">
            <GROUPRENDERER>

              <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
                <EXACT value="Freeway">
                  <SIMPLELINESYMBOL type="solid" width="9"
color="0,0,0" />
                </EXACT>
                <EXACT value="Street">
                  <SIMPLELINESYMBOL type="solid" width="8"
color="255,255,255" />
                </EXACT>
              </VALUEMAPRENDERER>

              <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
                <EXACT value="Freeway">
                  <SIMPLELINESYMBOL type="solid" width="7"
color="255,0,0" />
                </EXACT>
              </VALUEMAPRENDERER>

              <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
                <EXACT value="Freeway">
                  <SIMPLELINESYMBOL type="solid" width="3"
color="0,0,0" />
                </EXACT>
              </VALUEMAPRENDERER>

              <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
                <EXACT value="Freeway">
                  <SIMPLELINESYMBOL type="solid" width="1"
color="255,255,255" />
```

```
              </EXACT>
            </VALUEMAPRENDERER>

          </GROUPRENDERER>
        </SCALEDEPENDENTRENDERER>

        <SCALEDEPENDENTRENDERER upper="1:25000">

          <VALUEMAPLABELRENDERER lookupfield="ROAD_TYPE"
labelfield="NAME" linelabelposition="PlaceOnTop"
howmanylabels="One_label_per_name">
            <EXACT value="Freeway">
              <TEXTSYMBOL font="Arial" fontsize="14" fontstyle="bold"
fontcolor="0,0,255" glowing="153,153,153"/>
            </EXACT>
            <EXACT value="Street">
              <TEXTSYMBOL font="Arial" fontsize="14" fontstyle="bold"
fontcolor="0,0,0" />
            </EXACT>
          </VALUEMAPLABELRENDERER>

        </SCALEDEPENDENTRENDERER>

      </GROUPRENDERER>
    </LAYER>

    <LAYER type="featureclass" name="Roads" visible="true" id="5"
minscale="1:35000">
      <DATASET name="nyc_mainroads" type="line" workspace="shp_ws-0"
/>

      <GROUPRENDERER>
        <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
          <EXACT value="Freeway">
            <SIMPLELINESYMBOL type="solid" width="4" color="0,0,0" />
          </EXACT>
          <EXACT value="Secondary">
            <SIMPLELINESYMBOL type="solid" width="1" color="0,0,255"
/>
          </EXACT>
          <EXACT value="Primary">
            <SIMPLELINESYMBOL type="solid" width="1" color="255,0,0"
/>
          </EXACT>
        </VALUEMAPRENDERER>

        <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
          <EXACT value="Freeway">
            <SIMPLELINESYMBOL type="solid" width="2" color="255,0,0"
/>
          </EXACT>
        </VALUEMAPRENDERER>

      </GROUPRENDERER>
    </LAYER>
  </MAP>
 </CONFIG>
</ARCXML>
```

The following three figures show the New York City roads displayed at different scales. Depending on the scale, the street and freeway symbols change. The first map is at 1:200000, the second is at 1:30000, and the third is at 1:20000.



There are six layers in this map configuration file. Each of the layers has been assigned an alias name as shown in the following table. The alias names are what appear in the legend.

Layer Alias Name as Seen in Legend	Shapefile Name
Background	nyc_back
City Limits	nyc_bound
Parks	nyc_parks
Water	nyc_water
Roads	nyc_roads
Roads	nyc_mainroads

Nyc_back, nyc_bound, nyc_parks, nyc_water layers

Nyc_back, nyc_bound, nyc_parks, and nyc_water use the same SIMPLERENDERER format. The nyc_back layer is shown here as an example.

Nyc_back Layer:

```
      <LAYER type="featureclass" name="Background" visible="true"
id="0" >
        <DATASET name="nyc_back" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="145,239,254" filltype="solid"
/>
        </SIMPLERENDERER>
      </LAYER>
```

Nyc_mainroads

The nyc_mainroads and nyc_roads layers display at different scales, and only one of the two layers is included at any given scale. In the legend, both layers are labeled "Roads".

When the ArcIMS service is first loaded, nyc_mainroads displays. It continues to display until the scale is less than 1:35000. As a user zooms in to a scale of 1:35000, the nyc_mainroads layer turns off and the nyc_roads layer turns on.

The nyc_mainroads layer in the map configuration file is included below. First, the *minscale* in LAYER is set to 1:35000. This allows the layer to display only if the scale is greater than 1:35000. Second, for the rendering of this layer, a VALUEMAPRENDERER is used to divide the roads into Freeway, Secondary, and Primary roads based on the value in the field ROAD_TYPE. The road types are categorized using EXACT and are symbolized as follows:

Road Type	Symbol
Freeway	Uses a multipart line symbol of a red line with a width of two pixels on top of a black line with a width of four pixels.
Secondary	Uses a blue line symbol with a width of one pixel.
Primary	Uses a red line symbol with a width of one pixel.

The layer contains two VALUEMAPRENDERERs. The first VALUEMAPRENDERER includes Primary, Secondary, and the black line for Freeway. The second VALUEMAPRENDERER includes only the red line for the Freeway category. The VALUEMAPRENDERERs are grouped together by a GROUPRENDERER.

Nyc_mainroads layer:

```
      <LAYER type="featureclass" name="Roads" visible="true" id="5"
minscale="1:35000">
        <DATASET name="nyc_mainroads" type="line" workspace="shp_ws-0"
/>

        <GROUPRENDERER>
          <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
            <EXACT value="Freeway">
              <SIMPLELINESYMBOL type="solid" width="4" color="0,0,0" />
            </EXACT>
            <EXACT value="Secondary">
              <SIMPLELINESYMBOL type="solid" width="1" color="0,0,255"
/>
            </EXACT>
            <EXACT value="Primary">
              <SIMPLELINESYMBOL type="solid" width="1" color="255,0,0"
/>
            </EXACT>
          </VALUEMAPRENDERER>

          <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
            <EXACT value="Freeway">
              <SIMPLELINESYMBOL type="solid" width="2" color="255,0,0"
/>
            </EXACT>
          </VALUEMAPRENDERER>

        </GROUPRENDERER>
      </LAYER>
```

The figure below shows nyc_mainroads at a scale of 1:200000. Note that in the legend the layer is listed as "Roads".



Nyc_roads

The nyc_roads layer contains all the streets for New York City and the surrounding area. In the map configuration file, the LAYER *maxscale* is set to 1:35000 and does not turn on until a user zooms past a scale of 1:35000.

Using SIMPLERENDERER:

```
<LAYER type="featureclass" name="Roads" visible="true" id="4"
maxscale="1:35000">
<DATASET name="nyc_roads" type="line" workspace="shp_ws-0" />
...
</LAYER>
```

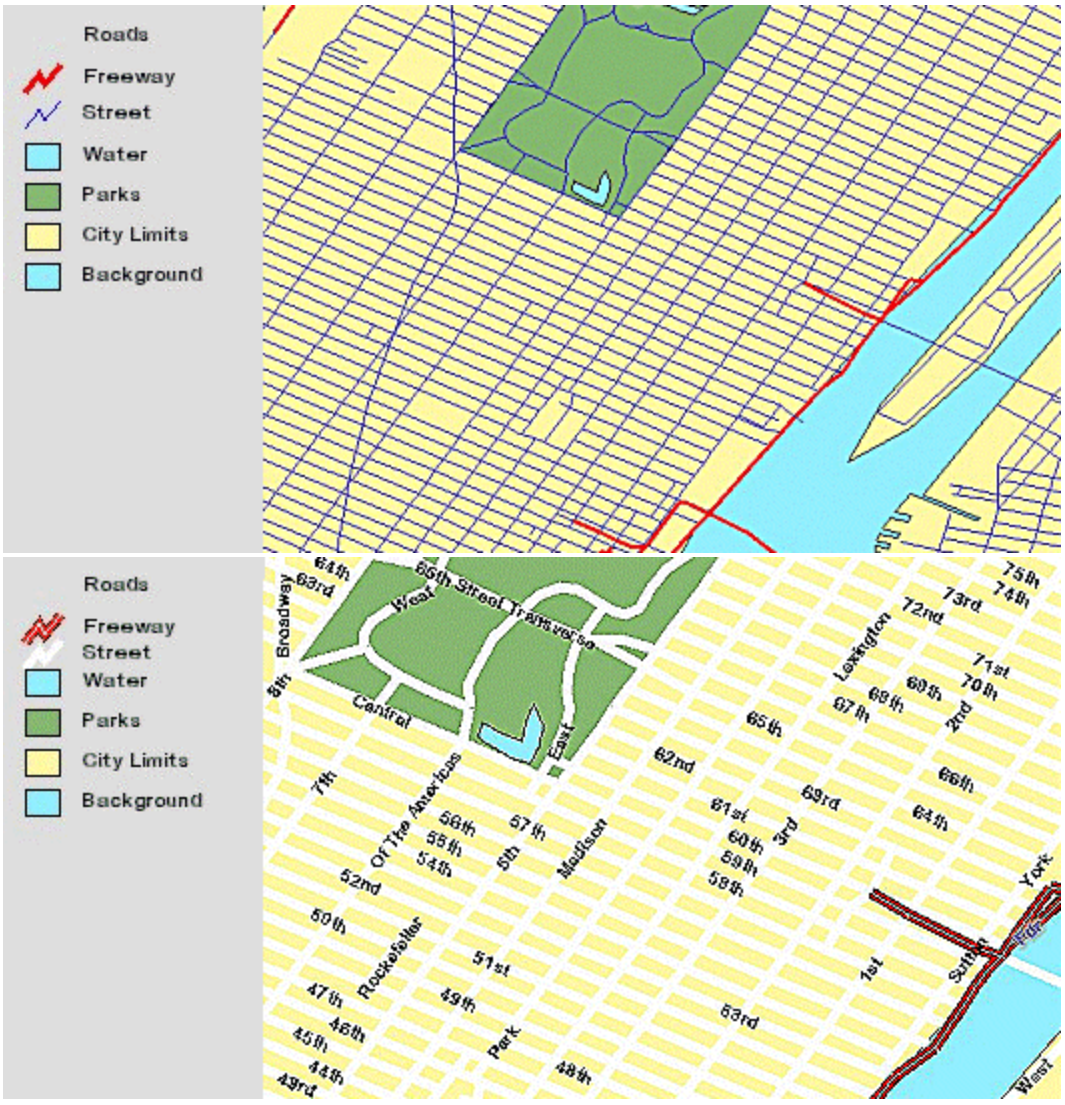
The nyc_roads layer is rendered two different ways depending on the current scale using SCALEDEPENDENTRENDERER. A broad overview of the use of SCALEDEPENDENTRENDERER in the layer is shown in the next example. At a scale between 1:21300 and 1:35000, the roads are rendered in a less detailed manner. When the scale is less than 1:21300, the road rendering becomes much more detailed. A third SCALEDEPENDENTRENDERER is used for labeling. The labels do not turn on until the scale is less than 1:25000. The SCALEDEPENDENTRENDERERs are grouped together inside a GROUPRENDERER.

Using SIMPLERENDERER:

```
    <LAYER…>
      <GROUPRENDERER>
        <SCALEDEPENDENTRENDERER lower="1:21300">
```

```
        …less detailed rendering…
      </SCALEDEPENDENTRENDERER>

      <SCALEDEPENDENTRENDERER upper="1:21300">
        …more detailed rendering…
      </SCALEDEPENDENTRENDERER>

      <SCALEDEPENDENTRENDERER upper="1:25000">
        …label rendering…
      </SCALEDEPENDENTRENDERER>
    </GROUPRENDERER>
  </LAYER>
```

The following figures show a snapshot of *nyc_roads* at two scales. The scale for the first map is 1:30000, and the roads are less detailed. The scale for the second map is 1:20000, showing more detail in the roads.



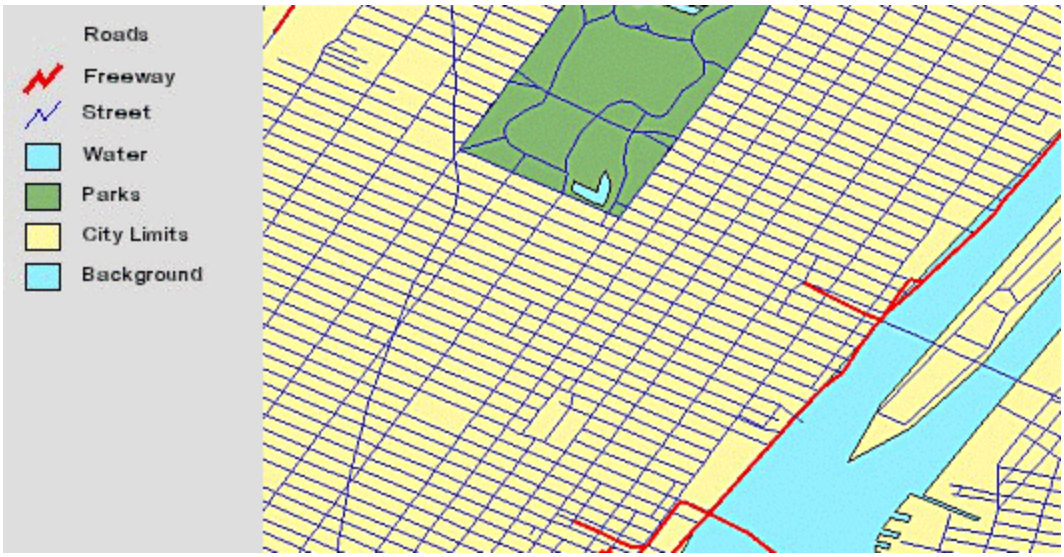
The first SCALEDEPENDENTRENDERER with the less detailed rendering has a minimum scale set to 1:21300 (*lower="1:21300"*). The VALUEMAPRENDERER

divides the roads into two categories based on the value in the field ROAD_TYPE: Freeway and Street. The freeways are a thick red line, and the streets are a thin blue line.

Using SCALEDEPENDENTRENDERER for nyc_roads when the scale is greater than 1:21000:

```
          <SCALEDEPENDENTRENDERER lower="1:21300" >

            <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
              <EXACT value="Freeway">
                <SIMPLELINESYMBOL type="solid" width="3"
color="255,0,0" />
              </EXACT>
              <EXACT value="Street">
                <SIMPLELINESYMBOL type="solid" width="1"
color="0,0,255" />
              </EXACT>
            </VALUEMAPRENDERER>

          </SCALEDEPENDENTRENDERER>
```

The figure below shows the map at a scale of 1:30000.



When the scale passes 1:21300, the second SCALEDEPENDENTRENDERER with more detailed rendering becomes active. Inside this SCALEDEPENDENTRENDERER is a group of VALUEMAPRENDERERs.

- The first VALUEMAPRENDERER uses the two categories of Street and Freeway. The streets are a thick solid white line eight pixels wide. The freeways are a multipart line symbol. The base level of the Freeway is included inside this renderer and is a black line nine pixels wide.
- The remaining three VALUEMAPRENDERERs create the rest of the multipart freeway symbol: red as the second layer seven pixels wide, black as the third layer three pixels wide, and white as the top layer one pixel wide.

Using SCALEDEPENDENTRENDERER for nyc_roads when the scale is less than 1:21000:

```
          <SCALEDEPENDENTRENDERER upper="1:21300">
            <GROUPRENDERER>

              <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
                <EXACT value="Freeway">
                  <!--black-->
                  <SIMPLELINESYMBOL type="solid" width="9"
color="0,0,0" />
                </EXACT>
                <EXACT value="Street">
                  <SIMPLELINESYMBOL type="solid" width="8"
color="255,255,255" />
                </EXACT>
              </VALUEMAPRENDERER>

              <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
                <EXACT value="Freeway">
                  <!--red-->
                  <SIMPLELINESYMBOL type="solid" width="7"
color="255,0,0" />
                </EXACT>
              </VALUEMAPRENDERER>

              <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
                <EXACT value="Freeway">
                  <!--black-->
                  <SIMPLELINESYMBOL type="solid" width="3"
color="0,0,0" />
                </EXACT>
              </VALUEMAPRENDERER>

              <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
                <EXACT value="Freeway">
                  <!--white-->
                  <SIMPLELINESYMBOL type="solid" width="1"
color="255,255,255" />
                </EXACT>
              </VALUEMAPRENDERER>

            </GROUPRENDERER>
          </SCALEDEPENDENTRENDERER>
```

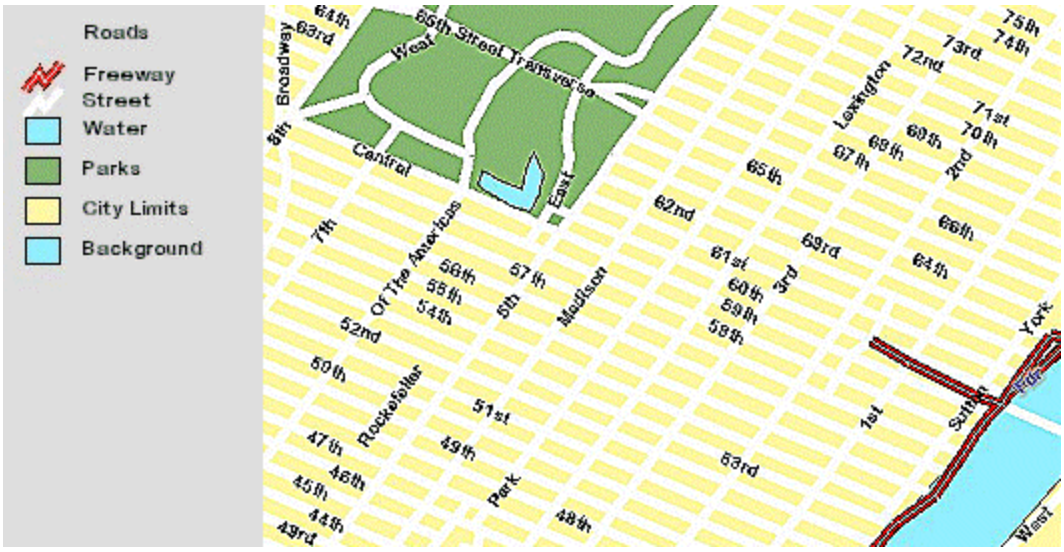
The third SCALEDEPENDENTRENDERER is used to set the scale for labeling. In this layer, VALUEMAPLABELRENDERER becomes active when the scale is less than 1:25000. The streets and freeways both use TEXTSYMBOL, but the instructions for drawing the labels are different. Freeways use a bold Arial font with glow. Streets use a bold Arial font without the glow.

Labeling for nyc_roads:

```
          <SCALEDEPENDENTRENDERER upper="1:25000">

            <VALUEMAPLABELRENDERER lookupfield="ROAD_TYPE"
```

```
labelfield="NAME" linelabelposition="PlaceOnTop"
howmanylabels="One_label_per_name">
              <EXACT value="Freeway">
                <TEXTSYMBOL font="Arial" fontsize="14" fontstyle="bold"
fontcolor="0,0,255" glowing="153,153,153"/>
              </EXACT>
              <EXACT value="Street">
                <TEXTSYMBOL font="Arial" fontsize="14" fontstyle="bold"
fontcolor="0,0,0" />
              </EXACT>
            </VALUEMAPLABELRENDERER>

          </SCALEDEPENDENTRENDERER>
```

The following figure shows the detailed streets and freeways at a scale of 1:20000. At this scale, the labels are also included.



This case study shows different ways to use the ArcXML renderers. In summary, the basic rules for using these renderers are:

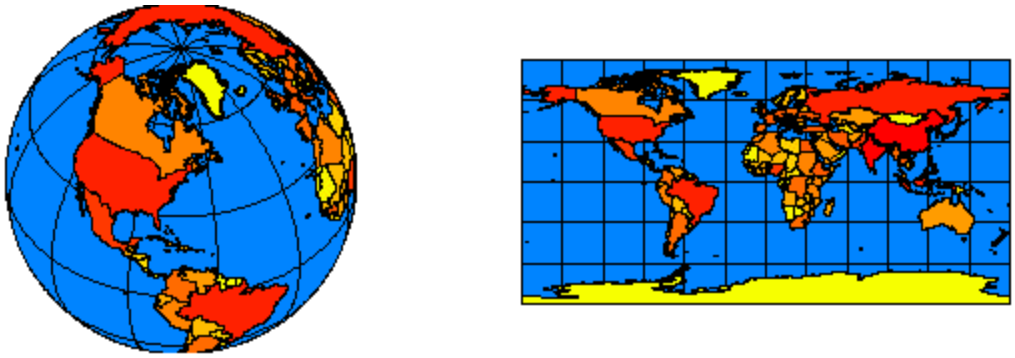
- Only one symbol can be used inside a renderer.
- SIMPLERENDERER and SIMPLELABELRENDERER are used when an entire layer is symbolized or labeled the same way.
- VALUEMAPRENDERER and VALUEMAPLABELRENDERER are used when different categories based on a value in a field are used to symbolize and label a layer.
- Two or more renderers are grouped together using GROUPRENDERER.
- Multiple GROUPRENDERERs can exist within other GROUPRENDERERs.
- SCALEDEPENDENTRENDERER is used to set scales for changing a layer's symbolization and labeling.

Using Projection Elements

Introduction

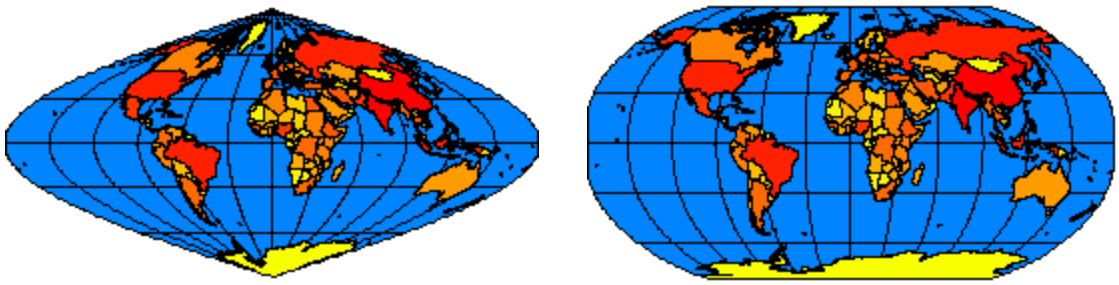
Features on a map refer to the actual locations of the objects they represent in the real world. The positions of objects on the earth's spherical surface are measured in degrees of latitude and longitude, also known as *geographic coordinates*. Though latitude and longitude can locate exact positions on the surface of the earth, they are not uniform units of measure; only along the equator does the distance represented by one degree of longitude approximate the distance represented by one degree of latitude. To overcome measurement difficulties, data is often transformed from the three-dimensional geographic coordinate system to the two-dimensional planar surface in a *projected coordinate system*. Projected coordinate systems describe the distance from an origin (0,0) along two separate axes: a horizontal x-axis representing east-west and a vertical y-axis representing north-south.

Because the earth is round and maps are flat, getting information from the curved surface to a flat one involves a mathematical formula call a *map projection*. A map projection transforms latitude and logitude to x,y coordinates in a projected coordinate system.



Locations are expressed as latitude and longitude on the globe and as x,y coordinates on a map.

This process of flattening the earth causes distortions in one or more of the following spatial properties: distance, area, shape, and direction. No projection can preserve all these properties, and as a result, all flat maps are distorted to some degree. Fortunately, you can choose from many different map projections. Each is distinguished by its suitability for representing a particular portion and amount of the earth's surface and by its ability to preserve distance, area, shape, or direction. Some map projections minimize distortion in one property at the expense of another, while others strive to balance the overall distortion. As a mapmaker, you can decide which properties are most important and choose a projection that suits your needs.



Displaying the world using the Sinusoidal projection (left) and the Robinson projection (right).

Reasons for using a projected coordinate system

- You want to make accurate measurements from your map and be sure that spatial analysis options calculate distance correctly. Latitude/Longitude is a good system for storing spatial data but not very good for viewing, querying, or analyzing maps. Degrees of latitude and longitude are not consistent units of measure for area, shape, distance, and direction.
- You are making a map in which you want to preserve one or more of these properties: area, shape, distance, and direction.
- You are making a small-scale map such as a national or world map. With a small-scale map, your choice of map projection determines the overall appearance of the map. For example, with some projections, lines of latitude and longitude will appear curved; with others they will appear straight.
- Your organization mandates using a particular projected coordinate system for all maps.

Choosing a map projection

Here are a few things to consider when choosing a projection:

- Which spatial properties do you want to preserve?
- Where is the area you're mapping? Is your data in a polar region? An equatorial region?
- What shape is the area you're mapping? Is it square? Is it wider in the east-west direction?
- How big is the area you're mapping? On large-scale maps, such as street maps, distortion may be negligible because your map covers only a small part of the earth's surface. On small-scale maps, where a small distance on the map represents a considerable distance on the earth, distortion may have a bigger impact, especially if you use your map to compare or measure shape, area, or distance.

Answering these questions will determine what map projection and thus what projected coordinate system you'll want to use to display your data.

Map projections can be generally classified according to what spatial attribute they preserve:

- *Equal Area* projections preserve area. Many thematic maps use an equal area projection. Maps of the United States commonly use the Albers Equal Area Conic projection.
- *Conformal* projections preserve shape and area useful for navigational charts and weather maps. Shape is preserved for small areas, but the shape of a large area such as a continent will be significantly distorted. The Lambert Conformal Conic and Mercator projections are common conformal projections.
- *Equidistant* projections preserve distance but no projection can preserve distances from all points to all other points. Instead, distance can be held true from one point (or a few points) to all other points or along all meridians or parallels. If you will be using your map to find features that are within a certain distance of other features, you should use an equidistant map projection.
- *Azimuthal* projections preserve direction from one point to all other points. This quality can be combined with equal area, conformal, and equidistant projections, as in the Lambert Equal Area Azimuthal and the azimuthal equidistant projections.
- Other projections minimize overall distortion but don't preserve any of the four spatial properties of area, shape, distance, and direction. The Robinson projection, for example, is neither equal area nor conformal but is aesthetically pleasing and useful for general mapping.

Projection Elements in ArcIMS

Projections in ArcIMS are handled by the ArcIMS Spatial Server. Three projection elements are used to retrieve data in the correct projected coordinate system:

- COORDSYS
- FEATURECOORDSYS
- FILTERCOORDSYS

With ArcIMS, the term *coordinate system*, which includes both geographic and projected coordinate systems, is used to describe the information about the projection, as well as other specifics such as datum, units, and meridians.

Each projection element defines a coordinate system by either an ID or a definition string. For a complete list of supported IDs and definition strings, see:

- Projected Coordinate Systems Listing [Sorted by projection ID] [Sorted by name]
- Geographic Coordinate Systems Listing [Sorted by projection ID] [Sorted by name]
- Datum Transformation Listing [Sorted by projection ID] [Sorted by name]

COORDSYS

COORDSYS defines the projection metadata for a data layer. It does not reproject the data to another coordinate system; it merely states what projection that layer is in. COORDSYS can be used with:

- Vector data including shapefiles, coverages in ArcSDE for Coverages, and ArcSDE layers. With vector data, if COORDSYS is not included with the layer, the ArcIMS Spatial Server reads any *.prj file associated with a shapefile or coverage or the spatial references table in ArcSDE.
- Imagery including images in a directory and images in ArcSDE.
- Acetate layers when the attribute *units="database"* is used in OBJECT.

If no *.prj file or spatial references table or COORDSYS is present, the layer is not projected. If a layer does not project as expected, double-check whether the layer has a *.prj file or spatial reference table or COORDSYS. For ArcMap Image Services, the coordinate system for each layer is handled in the ArcMap document (*.mxd or *.pmf).

FEATURECOORDSYS

FEATURECOORDSYS is used to specify a common coordinate system in an ArcIMS service to which each layer should be transformed. In an ArcIMS service, FEATURECOORDSYS represents the overall projection and can be thought of as the default coordinate system. This coordinate system can be different than the coordinate system of any of the layers.

Requests with no projection elements made to an ArcIMS service will receive data in the default coordinate system of the service. A request can override the service FEATURECOORDSYS with its own FEATURECOORDSYS. For example, if a service is in Robinson and a request includes a FEATURECOORDSYS of Mollweide, then the ArcIMS Spatial Server returns the data in Mollweide.

When FEATURECOORDSYS is present, the map units are calculated automatically by the ArcIMS Spatial Server. If MAPUNITS is present, it is ignored. Having the correct map units is important to make sure scale dependencies in LAYER, SCALEDEPENDENTRENDERER, and OBJECT are correctly calculated.

FILTERCOORDSYS

FILTERCOORDSYS is used to specify the coordinate system of the requesting client. Much like COORDSYS, FILTERCOORDSYS can be thought of as metadata, this time for the client. In the request, coordinates in the extents and spatial filters are in the coordinate system of the FILTERCOORDSYS.

FILTERCOORDSYS must be included in a map configuration file if FEATURECOORDSYS is present. As with requests, FILTERCOORDSYS represents the coordinates in the extents and spatial filters of the map configuration file. If these

coordinates are different than the overall FEATURECOORDSYS, then FILTERCOORDSYS will have a different ID or definition string.

Using the Projection Elements

This next section discusses different scenarios for using the projection elements. One important note: the ArcIMS clients do not support projections. All projections must be handled by the ArcIMS Spatial Server. In addition, when using the ArcIMS viewers, FILTERCOORDSYS and FEATURECOORDSYS must have the same ID. The ArcIMS HTML Viewer can be customized to accommodate different IDs for FEATURECOORDSYS and FILTERCOORDSYS. The ArcIMS Java Viewers cannot be customized this way.

Map configuration file without FILTERCOORDSYS and FEATURECOORDSYS

The example below shows a map configuration file that does not use FILTERCOORDSYS or FEATURECOORDSYS but does use COORDSYS for the layers. The two layers are both world shapefile layers. One is in World Mollweide (id="54009") and the other is in World Robinson (id="54030"). Remember, if a shapefile has a *.prj file associated with it, COORDSYS does not need to be included. If neither a *.prj file nor COORDSYS is present, the layer is not projected.

Using COORDSYS in a map configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-14628440.851850007" miny="-9020047.848073646"
maxx="15705351.712200116" maxy="8748562.401522137"
name="Initial_Extent" />
        <MAPUNITS units="meters" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-12" directory="c:\data" />
      </WORKSPACES>
      <LAYER type="featureclass" name="Ocean" visible="true" id="22">
        <DATASET name="WORLD30" type="polygon" workspace="shp_ws-12" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="240,255,255" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Robinson" visible="true" id="0">
        <DATASET name="cntry94_Robinson" type="polygon"
workspace="shp_ws-12" />
```

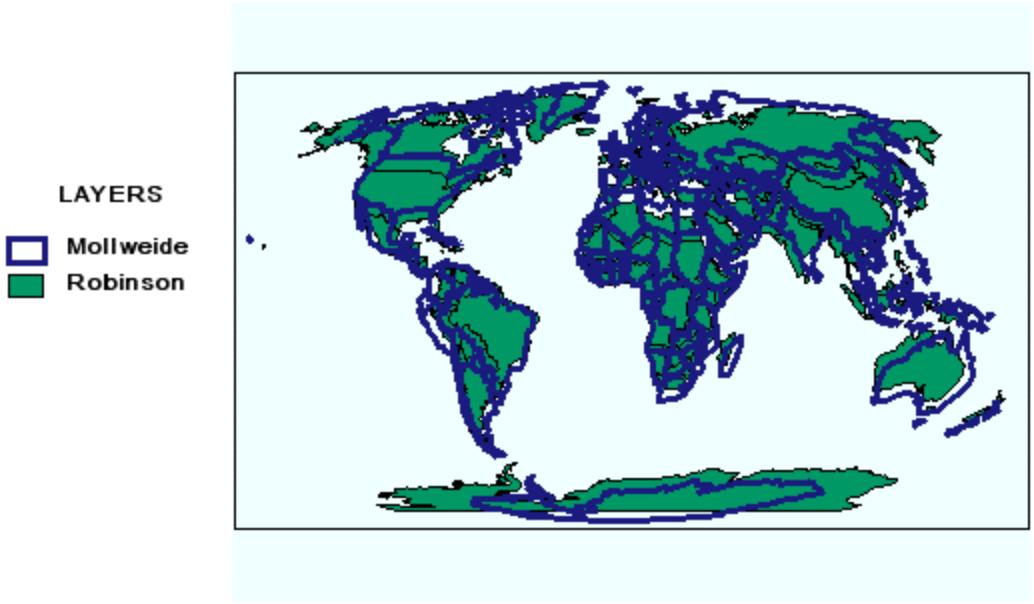
```
        <COORDSYS id="54030" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="0,153,102"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Mollweide" visible="true"
id="1">
        <DATASET name="Cntry94_Mollweide" type="polygon"
workspace="shp_ws-12" />
        <COORDSYS id="54009" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltransparency="0.0" boundarywidth="3"
boundarycolor="27,27,127" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

To view a map, a GET_IMAGE request can be made to this Image Service. The following request includes no projection elements.

GET_IMAGE request with no projection elements:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

In the returned image, the two layers do not overlay correctly because they are not in the same coordinate system. Remember, the ArcIMS viewers cannot project data.

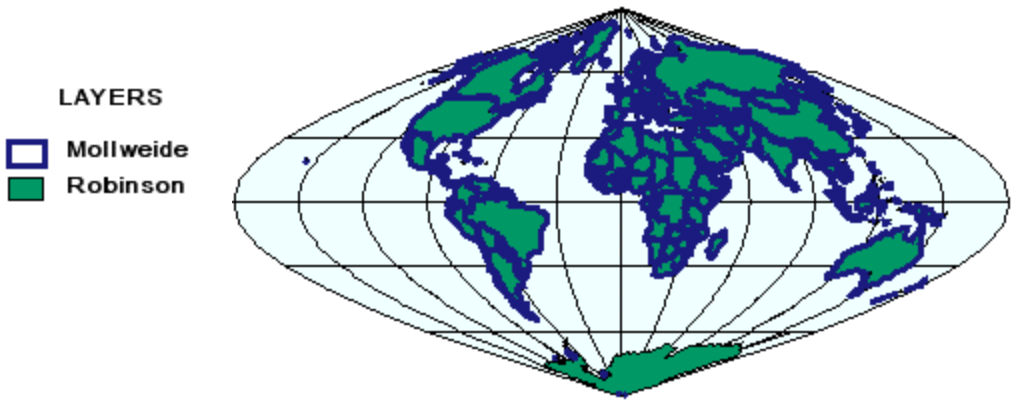


In the next GET_IMAGE request, FEATURECOORDSYS and FILTERCOORDSYS are included and are set to World Sinusoidal, which has an *id="54008"*.

GET_IMAGE request using FILTERCOORDSYS and FEATURECOORDSYS:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <FILTERCOORDSYS id="54008" />
        <FEATURECOORDSYS id="54008" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

In the returned image, the two layers are aligned in the Sinusoidal projection.



The last GET_IMAGE example worked correctly because the two layers have coordinate systems with similar minimum and maximum X and Y extents. In reality, an ENVELOPE is normally included in the request. Coordinates used in ENVELOPE and inside SPATIALFILTER should be in the coordinate system of FILTERCOORDSYS. In the next example, an ENVELOPE is included in the request. The map units are calculated by the ArcIMS Spatial Server based on the ID selected for FEATURECOORDSYS. Hence, MAPUNITS is not required in requests.

GET_IMAGE request with ENVELOPE:

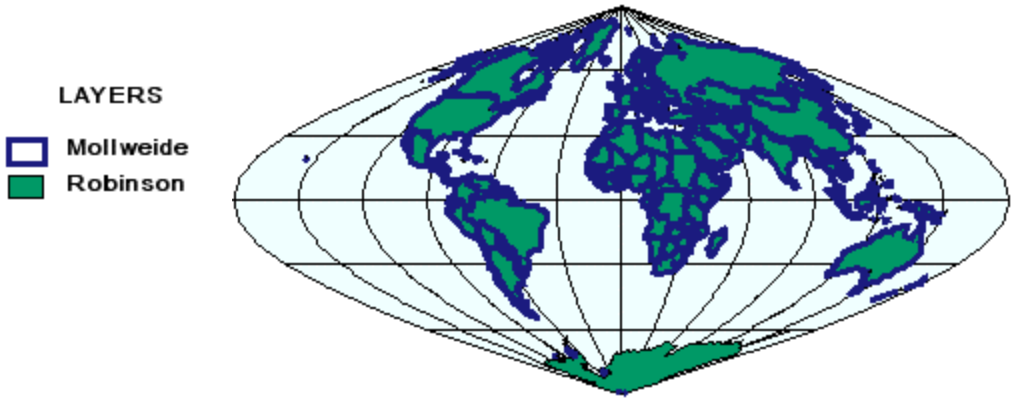
```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-14628440.851850007" miny="-9020047.848073646"
maxx="15705351.712200116" maxy="8748562.401522137" />
        <FILTERCOORDSYS id="54008" />
        <FEATURECOORDSYS id="54008" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

With the above request, the returned ENVELOPE is guaranteed to be correct.

IMAGE response with ENVELOPE:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-14628440.85185" miny="-11510914.9347945"
maxx="15705351.7122001" maxy="11239429.488243" />
      <OUTPUT file="C:\ArcIMS\output\proj_paper_MYCOMPUTER79217360.jpg"
url="http://mycomputer.esri.com/output/proj_paper_MYCOMPUTER79217360.jpg"
/>
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

The returned image is the same as the previous image.



Map configuration file with FEATURECOORDSYS and FILTERCOORDSYS

FILTERCOORDSYS and FEATURECOORDSYS can be included in the map configuration file as shown in the next example. For both elements, the coordinate system ID is set to World Sinusoidal (*id="54008"*). Although MAPUNITS is included, it is ignored when FEATURECOORDSYS is present.

Using FILTERCOORDSYS AND FEATURECOORDSYS in a map configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-14628440.851850007" miny="-9020047.848073646"
maxx="15705351.712200116" maxy="8748562.401522137"
name="Initial_Extent" />
        <MAPUNITS units="meters" />
        <FEATURECOORDSYS id="54008" />
        <FILTERCOORDSYS id="54008" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-12" directory="c:\data" />
      </WORKSPACES>
      <LAYER type="featureclass" name="Ocean" visible="true" id="22">
        <DATASET name="WORLD30" type="polygon" workspace="shp_ws-12" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="240,255,255" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Robinson" visible="true" id="0">
        <DATASET name="cntry94_Robinson" type="polygon"
workspace="shp_ws-12" />
        <COORDSYS id="54030" />
```

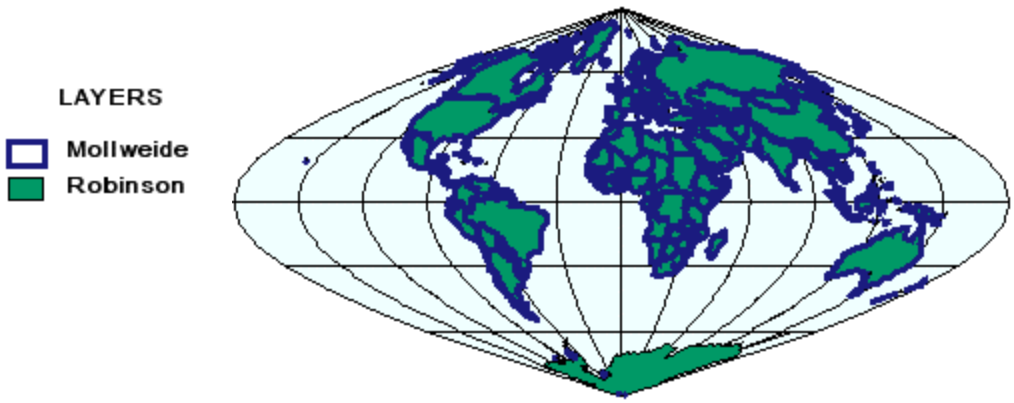
```
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="0,153,102"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Mollweide" visible="true"
id="1">
        <DATASET name="Cntry94_Mollweide" type="polygon"
workspace="shp_ws-12" />
        <COORDSYS id="54009" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltransparency="0.0" boundarywidth="3"
boundarycolor="27,27,127" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Once again, a GET_IMAGE request that includes no projection elements can be made to the service.

GET_IMAGE request with no projection elements:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

This time in the returned image, the two layers overlay in the Sinusoidal projection, the default from the service.



Regardless of the coordinate system of the service, a request can always include projection elements that override the projection elements in the service. If you want to

know the projection of a service you are accessing, you can send a GET_SERVICE_INFO request to find out which FEATURECOORDSYS and FILTERCOORDSYS are in use.

GET_SERVICE_INFO request:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_SERVICE_INFO fields="false" envelope="false" renderer="false"
extensions="false" />
  </REQUEST>
</ARCXML>
```

The response includes FILTERCOORDSYS and FEATURECOORDSYS if they are present in the ArcIMS service.

SERVICEINFO response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <IMAGELIMIT pixelcount="2250000"/>
      </ENVIRONMENT>
      <PROPERTIES>
        <FEATURECOORDSYS id="54008" />
        <FILTERCOORDSYS id="54008" />
        <ENVELOPE minx="-14628440.85185" miny="-9020047.84807364"
maxx="15705351.7122001" maxy="8748562.40152213" name="Initial_Extent"
/>
        <MAPUNITS units="meters" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="Ocean"
id="22">
        <FCLASS type="polygon"></FCLASS>
      </LAYERINFO>
      <LAYERINFO type="featureclass" visible="true" name="Robinson"
id="0">
        <FCLASS type="polygon"></FCLASS>
      </LAYERINFO>
      <LAYERINFO type="featureclass" visible="true" name="Mollweide"
id="1">
        <FCLASS type="polygon"></FCLASS>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

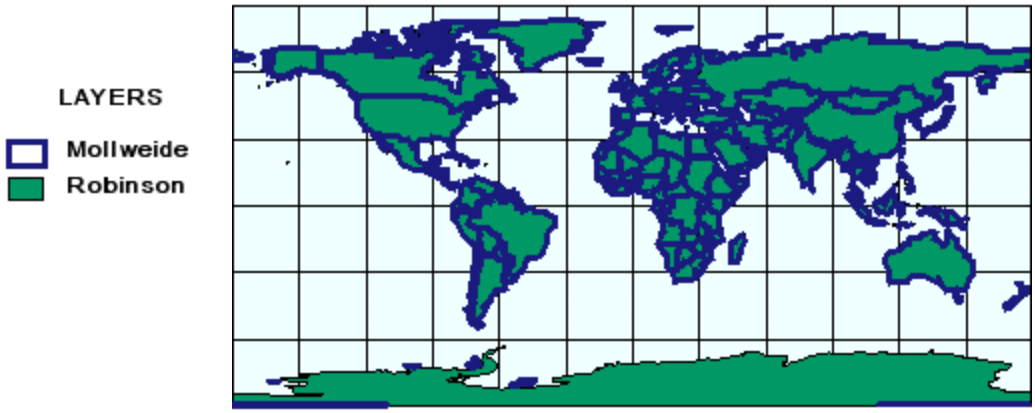
Once you know the projection information of an ArcIMS service, you can override FEATURECOORDSYS and FILTERCOORDSYS in a request if necessary. In the next GET_IMAGE request, FEATURECOORDSYS and FILTERCOORDSYS are both set to

geographic coordinates with an id="4326". Since the client is in geographic coordinates (represented by FILTERCOORDSYS), the ENVELOPE needs to be in geographic coordinates.

GET_IMAGE request using FILTERCOORDSYS and FEATURECOORDSYS:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <FILTERCOORDSYS id="4326" />
        <FEATURECOORDSYS id="4326" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

In the returned image, the two layers are aligned in geographic coordinates.



Changing projections in a customized ArcIMS HTML Viewer

Assume now that your ArcIMS HTML Viewer is currently in geographic coordinates ("4326"), but you want a map returned in Robinson ("54030"). In this scenario, FILTERCOORDSYS, the current projection of the client, is set to *id="4326"*. Coordinates in the request, such as those in ENVELOPE, should all be in geographic coordinates.

FEATURECOORDSYS defines the coordinate system of the map returned to the client. In this case, FEATURECOORDSYS is set to *id="54030"*.

GET_IMAGE request using FILTERCOORDSYS and FEATURECOORDSYS with ENVELOPE in decimal degrees:

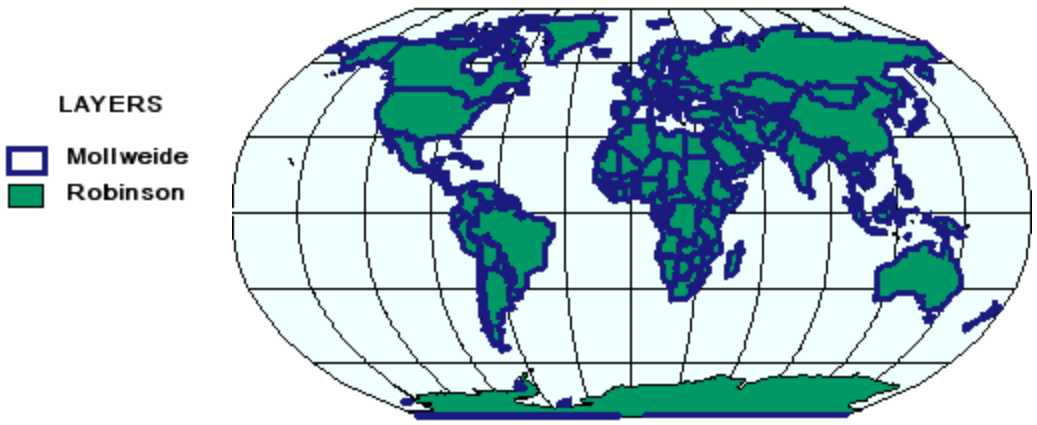
```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <FILTERCOORDSYS id="4326" />
        <FEATURECOORDSYS id="54030" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

In the response, the ENVELOPE is in Robinson coordinates.

IMAGE response with ENVELOPE in Robinson coordinates:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-16986727.8564406" miny="-12740045.8923304"
maxx="16986727.8564406" maxy="12740045.8923304" />
      <OUTPUT file="C:\ArcIMS\output\proj_paper_MYCOMPUTER79217360.jpg"
url="http://mycomputer.esri.com/output/proj_paper_MYCOMPUTER79217360.jpg"
/>
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

In the returned image, the two layers are now in Robinson.



Defining a Coordinate System

A coordinate system can be defined in two ways: using a predefined coordinate system ID or a coordinate system definition string. The predefined IDs are a quick way to reference a coordinate system by using a single reference number. The coordinate system definition string uses a string to describe all parameters for a projection. The advantage of using the definition string is that parameters in the string can be modified.

For a complete list of supported IDs and definition strings, see:

- Projected Coordinate Systems Listing
- Geographic Coordinate Systems Listing
- Datum Transformation Listing

As an example, the coordinate system ID for World Robinson is 54030.

The coordinate system definition string for World Robinson is:

World Robinson,54030
PROJCS["World_Robinson",GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",0.0174532925199433295]],PROJECTION["Robinson"],PARAMETER["False_Easting",0],PARAMETER["False_Northing",0],PARAMETER["Central_Meridian",0],UNIT["Meter",1]]

In the examples so far, only the IDs have been used. These IDs can be replaced by the definition string. In the next example, an Image Service includes the definition string for World Robinson for FEATURECOORDSYS and FILTERCOORDSYS. Note that the quotation marks (") in the definition string must be escaped with **"**.

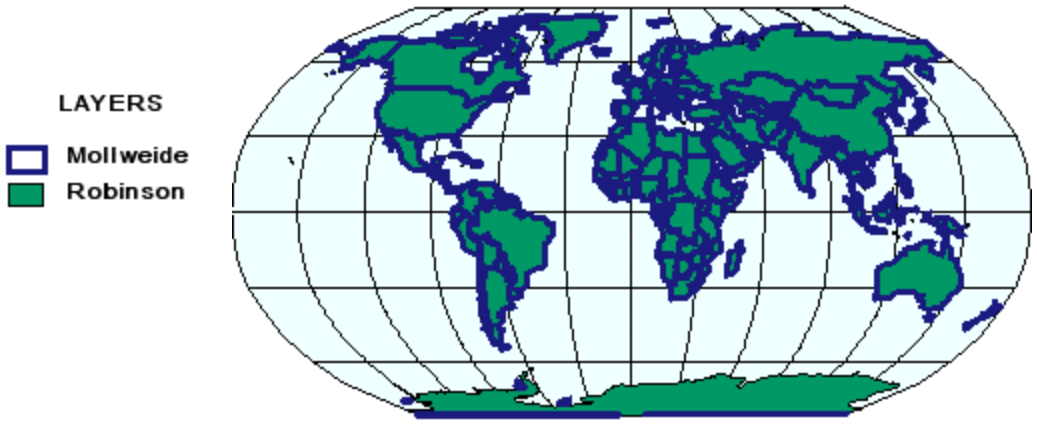
PROJCS[**&ampquot;**World_Robinson**&ampquot;**,GEOGCS[**&ampquot;**GCS_WGS_1984**&ampquot;**,DATUM[**&ampquot;**D_WGS_1984**&ampquot;**,SPHEROID[**&ampquot;**WGS_1984**&ampquot;**,6378137,298.257223563]],PRIMEM[**&ampquot;**Greenwich**&ampquot;**,0],UNIT[&ampquot;Degree**&ampquot;**,0.0174532925199433295]],PROJECTION[**&ampquot;**Robinson**&ampquot;**],PARAMETER[**&ampquot;**False_Easting**&ampquot;**,0],PARAMETER[**&ampquot;**False_Northing**&ampquot;**,0],PARAMETER[**&ampquot;**Central_Meridian**&ampquot;**,0],UNIT[**&ampquot;**Meter**&ampquot;**,1]]

Using a projection definition string for FILTERCOORDSYS AND FEATURECOORDSYS in a map configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
```

```
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-14628440.851850007" miny="-9020047.848073646"
maxx="15705351.712200116" maxy="8748562.401522137"
name="Initial_Extent" />
        <MAPUNITS units="meters" />
        <FEATURECOORDSYS
string="PROJCS[&ampquot;World_Robinson&ampquot;,GEOGCS[&ampquot;GCS_WGS
_1984&ampquot;,DATUM[&ampquot;D_WGS_1984&ampquot;,SPHEROID[&ampquot;WGS
_1984&ampquot;,6378137,298.257223563]],PRIMEM[&ampquot;Greenwich&ampquo
t;,0],UNIT[&ampquot;Degree&ampquot;,0.0174532925199433295]],PROJECTION[&
ampquot;Robinson&ampquot;],PARAMETER[&ampquot;False_Easting&ampquot;,0]
,PARAMETER[&ampquot;False_Northing&ampquot;,0],PARAMETER[&ampquot;Centr
al_Meridian&ampquot;,0],UNIT[&ampquot;Meter&ampquot;,1]]" />
        <FILTERCOORDSYS
string="PROJCS[&ampquot;World_Robinson&ampquot;,GEOGCS[&ampquot;GCS_WGS
_1984&ampquot;,DATUM[&ampquot;D_WGS_1984&ampquot;,SPHEROID[&ampquot;WGS
_1984&ampquot;,6378137,298.257223563]],PRIMEM[&ampquot;Greenwich&ampquo
t;,0],UNIT[&ampquot;Degree&ampquot;,0.0174532925199433295]],PROJECTION[&
ampquot;Robinson&ampquot;],PARAMETER[&ampquot;False_Easting&ampquot;,0]
,PARAMETER[&ampquot;False_Northing&ampquot;,0],PARAMETER[&ampquot;Centr
al_Meridian&ampquot;,0],UNIT[&ampquot;Meter&ampquot;,1]]" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-12" directory="c:\data" />
      </WORKSPACES>
      <LAYER type="featureclass" name="Ocean" visible="true" id="22">
        <DATASET name="WORLD30" type="polygon" workspace="shp_ws-12" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="240,255,255" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Robinson" visible="true" id="0">
        <DATASET name="cntry94_Robinson" type="polygon"
workspace="shp_ws-12" />
        <COORDSYS id="54030" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="0,153,102"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Mollweide" visible="true"
id="1">
        <DATASET name="Cntry94_Mollweide" type="polygon"
workspace="shp_ws-12" />
        <COORDSYS id="54009" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltransparency="0.0" boundarywidth="3"
boundarycolor="27,27,127" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

If a GET_IMAGE request is made to this service, the returned image includes the two layers in Robinson.



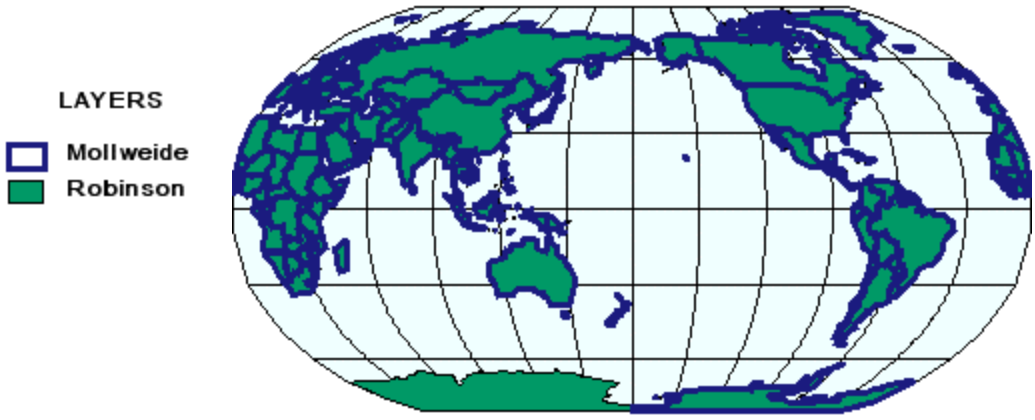
In the next example, an attribute in the definition string is changed. The central meridian for the FEATURECOORDSYS and FILTERCOORDSYS definition strings has been changed from 0 to 180.

Changing the central meridian in a projection definition string:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-14628440.851850007" miny="-9020047.848073646"
maxx="15705351.712200116" maxy="8748562.401522137"
name="Initial_Extent" />
        <MAPUNITS units="meters" />
        <FEATURECOORDSYS
string="PROJCS[&ampquot;World_Robinson&ampquot;,GEOGCS[&ampquot;GCS_WGS
_1984&ampquot;,DATUM[&ampquot;D_WGS_1984&ampquot;,SPHEROID[&ampquot;WGS
_1984&ampquot;,6378137,298.257223563]],PRIMEM[&ampquot;Greenwich&ampquo
t;,0],UNIT[&ampquot;Degree&ampquot;,0.0174532925199433295]],PROJECTION[&
ampquot;Robinson&ampquot;],PARAMETER[&ampquot;False_Easting&ampquot;,0]
,PARAMETER[&ampquot;False_Northing&ampquot;,0],PARAMETER[&ampquot;Centr
al_Meridian&ampquot;,180],UNIT[&ampquot;Meter&ampquot;,1]]" />
        <FILTERCOORDSYS
string="PROJCS[&ampquot;World_Robinson&ampquot;,GEOGCS[&ampquot;GCS_WGS
_1984&ampquot;,DATUM[&ampquot;D_WGS_1984&ampquot;,SPHEROID[&ampquot;WGS
_1984&ampquot;,6378137,298.257223563]],PRIMEM[&ampquot;Greenwich&ampquo
t;,0],UNIT[&ampquot;Degree&ampquot;,0.0174532925199433295]],PROJECTION[&
ampquot;Robinson&ampquot;],PARAMETER[&ampquot;False_Easting&ampquot;,0]
,PARAMETER[&ampquot;False_Northing&ampquot;,0],PARAMETER[&ampquot;Centr
al_Meridian&ampquot;,180],UNIT[&ampquot;Meter&ampquot;,1]]" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-12" directory="c:\data" />
      </WORKSPACES>
```

```
      <LAYER type="featureclass" name="Ocean" visible="true" id="22">
        <DATASET name="WORLD30" type="polygon" workspace="shp_ws-12" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="240,255,255" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Robinson" visible="true" id="0">
        <DATASET name="cntry94_Robinson" type="polygon"
workspace="shp_ws-12" />
        <COORDSYS id="54030" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="0,153,102"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Mollweide" visible="true"
id="1">
        <DATASET name="Cntry94_Mollweide" type="polygon"
workspace="shp_ws-12" />
        <COORDSYS id="54009" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltransparency="0.0" boundarywidth="3"
boundarycolor="27,27,127" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

If a GET_IMAGE request is made to this service, the returned image includes the two layers in Robinson centered on the Pacific Ocean.



Although the above examples with definition strings use map configuration files, definition strings can also be used in requests.

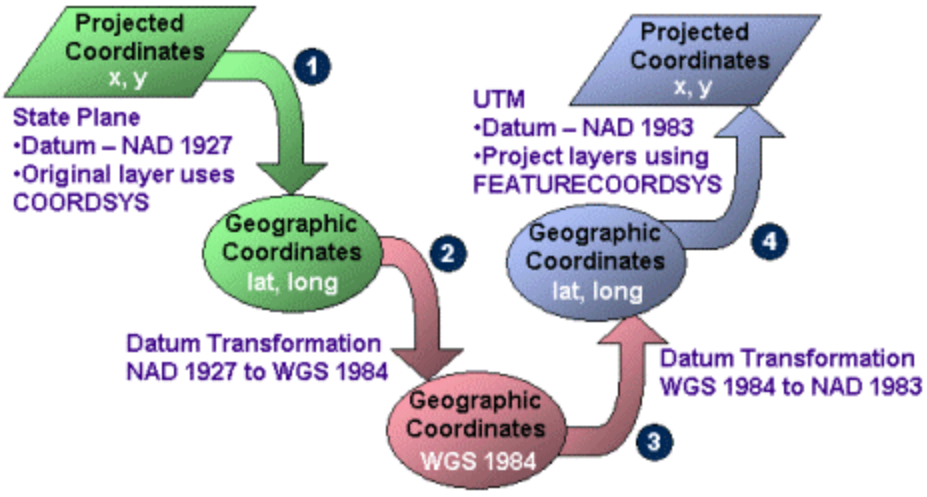
Using Datum Transformations

A datum is a reference frame defined by a spheroid and the spheroid's position relative to the center of the earth. A datum uses latitude and longitude or geographic coordinate systems to define the locations of points on the surface of a spheroid.

A datum transformation is a method that converts data between two geographic coordinate systems. Common geographic coordinate systems are WGS 1984 and NAD 1983. Only datum transformations to and from WGS 1984 are supported.

What happens behind the scenes during a datum transformation is a four-step process. In the following figure, the desired result is to transform a layer from State Plane coordinates in NAD 1927 to UTM coordinates in NAD 1983. All datum transformations take place to and from WGS 1984, so there are some intermediate steps.

- In Step 1, the original layer is in State Plane NAD 1927. This information should be included with COORDSYS within the layer. The projected data is transformed to geographic coordinates.
- In Step 2, the datum transformation is made from NAD 1927 to WGS 1984.
- In Step 3, the datum transformation is made from WGS 1984 to NAD 1983.
- In Step 4, the data is projected from geographic coordinates to the new projection, in this case UTM coordinates. The projection information is contained in FEATURECOORDSYS.



The attributes *datumtransformid* and *datumtransformstring* are used with the projection elements when datum transformation information needs to be included.

- When these attributes are used with COORDSYS and FILTERCOORDSYS, the datum transformation is from a non-WGS 1984 datum to WGS 1984. For example, Pulkovo_1942_To_WGS_1984 transforms data from Pulkovo 1942 to WGS 1984.

- When these attributes are used with FEATURECOORDSYS, the datum transformation is from WGS 1984 to a non-WGS 1984 datum. In the next example, the datum transformation is from WGS 1984 to Pulkovo 1942.

Transforming a layer in NAD 1927 (*datumtransformid="8070"*) State Plane coordinates (*id="26745"*) to NAD 1983 (*datumtransformid="8088"*) UTM coordinates (*id="26911"*):

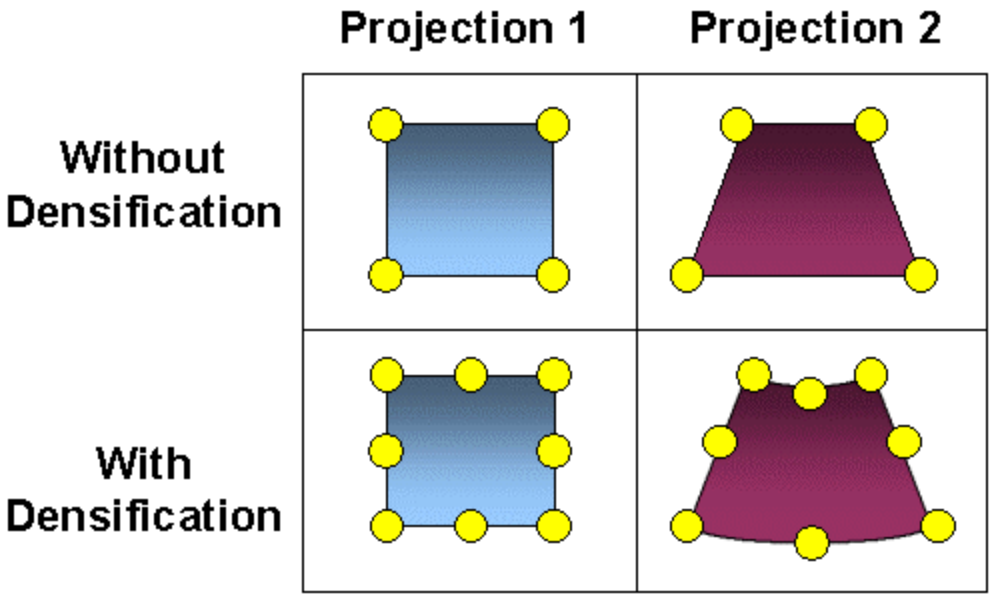
```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="51508.9214" miny="3429418.0119"
maxx="948491.0785" maxy="4102154.6297" name="Initial_Extent"
name="Initial_Extent" />
        <MAPUNITS units="feet" />
        <FEATURECOORDSYS id="26911" datumtransformid="8088" />
        <FILTERCOORDSYS id="26911" datumtransformid="8088" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-2" directory="<path to data>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="California" visible="true"
id="1">
        <DATASET name="calif" type="polygon" workspace="shp_ws-2" />
        <COORDSYS id="26745" datumtransformid="8070" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL color="0,0,255" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

The complete list of supported datum transformations is included in the Datum Transformation Listing.

Data Densification

The process of data densification adds points to a layer before the layer is projected. When data is projected, the line geometry changes. For example, a square in one projection may be trapezoidal in another projection. In many cases, the number of points defining the geometry in the original layer is not enough to describe the geometry correctly in the projected layer. In the following figure, without densification, the square

in Projection 1 is projected to the trapezoid as seen with Projection 2. When the data for Projection 1 is densified, the geometry for Projection 2 shows more detail.



For a layer in association with COORDSYS, DENSIFY can be used to define how much data should be densified before a layer is projected. If you find that features are missing from a layer in a requested image, usually around the periphery, try using DENSIFY on the layer. A word of caution about DENSIFY: if the *tolerance* value is too small, the ArcIMS Spatial Server can slow down considerably. It is better to start with a large number for *tolerance*. A guideline is to start with a value about 20 to 30 percent of the distance between the minimum and maximum x-coordinates of the current map extent.

Processing Time

Whenever the projection elements are used, the time to process a request on the ArcIMS Spatial Server can take considerably longer than when no projection elements are used. This is important when considering performance issues. If FILTERCOORDSYS and FEATURECOORDSYS are used, the amount of time to transform data can be an order of magnitude more than not using the elements depending on the complexity of the data. When using densification, it takes even longer to process the data depending on how much each layer has been densified. Ideally, if you have control over the data, all data and the service should be in the same projection and datum to minimize the need for using the projection elements.

Using GET_SERVICE_INFO and SERVICEINFO with Image and Feature Services

Introduction

GET_SERVICE_INFO is a request for accessing information about an ArcIMS service such as the environment, properties, and layer information. This information is returned in a SERVICEINFO response. This document covers Feature and Image Services. For information on ArcMap Image Services, see Using GET_SERVICE_INFO and SERVICEINFO with ArcMap Image Services.

The following map configuration file, started as an Image Service, is used for the GET_SERVICE_INFO examples in this document. The file consists of two layers from the ESRIDATA data set and an acetate layer. The following table summarizes the layer names, file name, file type and layer ID number.

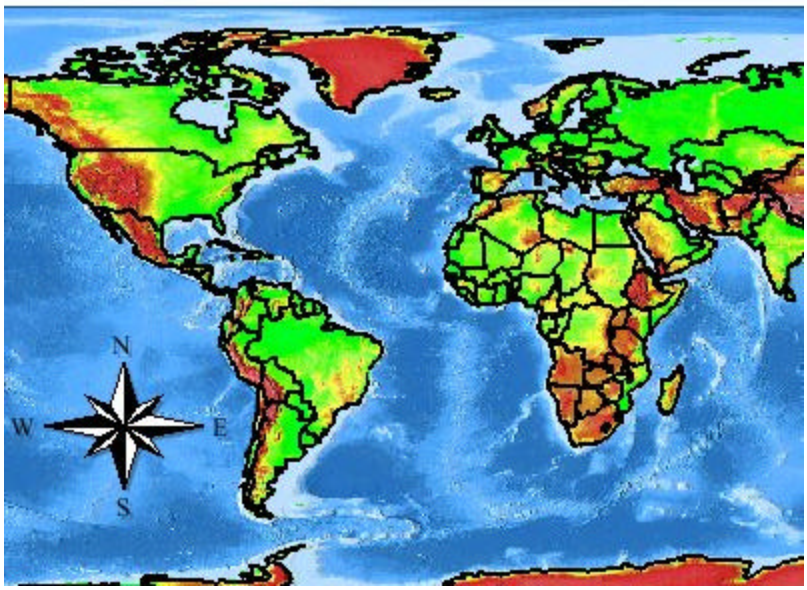
Layer Name	Data Name	Data Type	Layer ID
Background	WORLD_IMG.gif	Image	0
Countries	CNTRY94	Polygon	1
northarrow	None	Acetate	northarrow

Map configuration file used with the GET_SERVICE_INFO requests that follow:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <IMAGEWORKSPACE directory="<path to data>" name="jai_ws-4" />
        <SHAPEWORKSPACE name="shp_ws-5" directory="<path to WORLD
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="image" name="Background" visible="true" id="0">
        <DATASET name="WORLD_IMG.gif" type="image" workspace="jai_ws-4"
```

```
/>
      </LAYER>
      <LAYER type="featureclass" name="Countries" visible="true"
id="1">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-5" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltransparency="0.0"
boundarywidth="2"/>
        </SIMPLERENDERER>
        <EXTENSION type="Extract">
          <EXTRACTPARAMS clip="true">
            <OUTPUTFILE file="world">
              <OUTPUTFIELD name="NAME" alias="Country" />
            </OUTPUTFILE>
          </EXTRACTPARAMS>
        </EXTENSION>
        <EXTENSION type="Geocode">
          <GCSTYLE name="SingleField">
            <GCFIELD id="KeyField" name="NAME" />
          </GCSTYLE>
        </EXTENSION>
        <EXTENSION type="StoredQuery">
          <STOREDQUERIES>
            <STOREDQUERY name="Country">
              <QUERY where="( NAME = [%var%] )" subfields="#SHAPE# AREA
NAME ABBREVNAME FIPS_CODE WB_CNTRY" />
              <SQVAR position="0" name="[%var%]">
                <FIELD name="NAME" precision="0" type="12" size="40" />
              </SQVAR>
            </STOREDQUERY>
          </STOREDQUERIES>
        </EXTENSION>
      </LAYER>
      <LAYER type="acetate" name="northarrow" id="northarrow">
        <OBJECT units="pixel">
          <NORTHARROW type="6" size="40" coords="60 80"
shadow="32,32,32" angle="0" antialiasing="True" overlap="False"/>
        </OBJECT>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

The Background layer is an image of the world. The Countries layer contains country boundaries and includes three extensions: one for extract, one for geocode, and one for stored queries. The acetate layer consists of a north arrow. The map in the following figure is example output using this service.



GET_SERVICE_INFO Framework

A GET_SERVICE_INFO request requires only the GET_SERVICE_INFO element as shown in the example below.

GET_SERVICE_INFO request:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_SERVICE_INFO evelope="true" fields="true" renderer="true"
extensions="true" />
  </REQUEST>
</ARCXML>
```

SERVICEINFO Framework

Depending on which attributes are used with GET_SERVICE_INFO, all or a subset of information about the Image or Feature Service is returned in SERVICEINFO. By default, SERVICEINFO returns all information about a MapService. Based on the above request on the sample service, the following example shows the SERVICEINFO response.

SERVICEINFO response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
```

```
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="image" name="Background" visible="true" id="0">
        <ENVELOPE minx="-180" miny="-89.9747543334961"
maxx="179.9423828125" maxy="90" />
      </LAYERINFO>
      <LAYERINFO type="featureclass" visible="true" name="Countries"
id="1">
        <FCLASS type="polygon">
          <ENVELOPE minx="-180" miny="-90" maxx="180"
maxy="83.5960388183594" />
          <FIELD name="AREA" type="8" size="12" precision="3" />
          <FIELD name="NAME" type="12" size="40" precision="0" />
          <FIELD name="ABBREVNAME" type="12" size="12" precision="0" />
          <FIELD name="FIPS_CODE" type="12" size="2" precision="0" />
          <FIELD name="WB_CNTRY" type="12" size="3" precision="0" />
          <FIELD name="#SHAPE#" type="-98" size="0" precision="0" />
          <FIELD name="#ID#" type="-99" size="16" precision="0" />
        </FCLASS>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltransparency="0.0" boundarywidth="2"
/>
        </SIMPLERENDERER>
        <EXTENSION type="extract">
          <EXTRACTPARAMS clip="true">
            <OUTPUTFILE file="world">
              <OUTPUTFIELD name="NAME" alias="Country"/>
            </OUTPUTFILE>
          </EXTRACTPARAMS>
        </EXTENSION>
        <EXTENSION type="Geocode" >
          <GCSTYLE name="SingleField" />
        </EXTENSION>
        <EXTENSION type="StoredQuery">
          <STOREDQUERIES>
            <STOREDQUERY name="Country">
              <QUERY subfields="#SHAPE# AREA NAME ABBREVNAME FIPS_CODE
WB_CNTRY" where="( NAME = [%var%] )" >
              </QUERY>
              <SQVAR name="[%var%]" position="0">
                <FIELD name="NAME" type="12" precision="0" size="40" />
              </SQVAR>
            </STOREDQUERY>
          </STOREDQUERIES>
        </EXTENSION>
      </LAYERINFO>
      <LAYERINFO type="acetate" name="northarrow" visible="true"
id="northarrow"/>
```

```
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

SERVICEINFO is divided into three sections: ENVIRONMENT, PROPERTIES, and LAYERINFO.

ENVIRONMENT

ENVIRONMENT takes its values from the Image or Feature service or assigns default values. LOCALE and UIFONT are required, so the information in SERVICEINFO is always taken directly from the service. With SCREEN and SEPARATORS, the values are either those assigned in the service or default values assigned by the ArcIMS Spatial Server. For SCREEN, the default value for *dpi* is "96". For SEPARATORS, the default for *cs* (coordinate separator) and *ts* (tuple separator) are a space and semicolon, respectively.

IMAGELIMIT is always included and is set when an Image Service is started. By default, an image is limited in size to 4 MB, which corresponds to 1,048,576 pixels. Changes can be made to the image size and pixel count using the ArcIMS Administrator. For more information, see *Using ArcIMS*. The attribute *pixelcount* reports the pixel count from the ArcIMS Administrator, but it cannot be changed using GET_SERVICE_INFO.

CAPABILITIES is included only when forbidden elements are added to an access control list (ACL). The ACL file is used for user authentication to grant users access to specific Image or Feature Services. One way to limit access is to forbid one or more of the requests: GET_EXTRACT, GET_FEATURES, GET_GEOCODE, GET_IMAGE, GET_SERVICE_INFO, GET_LAYOUT, GET_RASTER_INFO, GET_METADATA, or PUBLISH_METADATA. For more information on forbidden elements, see "ArcIMS Servlet Connector" under the System Administration chapter in the ArcIMS Help.

In the example below, *forbidden* in CAPABILITIES is set to GET_GEOCODE and GET_EXTRACT. This means no requests with GET_GEOCODE or GET_EXTRACT will be processed.

ENVIRONMENT and its child elements:

```
<ENVIRONMENT>
  <LOCALE language="en" country="US" />
  <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
  <SEPARATORS cs=" " ts=";"/>
  <CAPABILITIES forbidden="GET_GEOCODE,GET_EXTRACT"/>
  <SCREEN dpi="96"/>
  <IMAGELIMIT pixelcount="1048576" />
</ENVIRONMENT>
```

PROPERTIES

PROPERTIES in SERVICEINFO repeats back all information in the PROPERTIES section of an Image or Feature Service. The overall ENVELOPE for the service is always included since it is required in a service. All other child elements are optional in a service and are included in the SERVICEINFO response only when included in the service. Optional child elements valid in both Feature and Image Services include MAPUNITS, FEATURECOORDSYS, and FILTERCOORDSYS. Optional child elements valid in Image Services only include BACKGROUND, LEGEND, IMAGESIZE, and OUTPUT.

PROPERTIES and its child elements:

```
<PROPERTIES>
  <FEATURECOORDSYS id="54030" />
  <FILTERCOORDSYS id="4326" />
  <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
  <MAPUNITS units="meters" />
  <LEGEND autoextend="true" backgroundcolor="255,255,0"/>
  <BACKGROUND color="0,0,255" />
  <IMAGESIZE width="480" height="640"/>
  <OUTPUT path="c:\arcims\website"
baseurl="http://mycomputer.domain.com/website"/>
</PROPERTIES>
```

LAYERINFO

LAYERINFO describes the layer information in a service and includes the layer type, name of the layer, whether the layer is visible, the layer ID, and the minimum and maximum scales at which the layer can be displayed. All these parameters are defined in the service LAYER. None of these parameters can be overridden in a request.

There are three LAYERINFO types, each of which corresponds to the same LAYER type in a map configuration file.

- **Image.** In addition to LAYERINFO, the child element ENVELOPE is always included with an image layer.
- **Acetate.** No child attributes are included in an acetate layer.
- **Featureclass.** In addition to LAYERINFO, featureclass layers always include the child element FCLASS, which describes whether the featureclass is polygon, line, or point. Also, as discussed in the next section, depending on different attributes selected in GET_SERVICE_INFO, additional information about the featureclass layer can be included in the SERVICEINFO response.

The following example shows each of the layer types with the minimum required information.

LAYERINFO and minimum returned information:

```
<LAYERINFO type="image" name="Background" visible="true" id="0"
  <ENVELOPE minx="-180" miny="-89.9747543334961" maxx="179.9423828125"
maxy="90" />
</LAYERINFO>
<LAYERINFO type="featureclass" visible="true" name="Countries" id="1">
  <FCLASS type="polygon"> </FCLASS>
</LAYERINFO>
<LAYERINFO type="acetate" name="northarrow" visible="true"
id="northarrow"/>
```

GET_SERVICE_INFO Attributes

GET_SERVICE_INFO has five optional attributes, four of which determine what information about a featureclass layer is returned in SERVICEINFO.

- *envelope*
- *fields*
- *renderer*
- *extensions*

The fifth attribute is *dpi*, which is discussed later in this session.

If the above attributes are set to "false", the minimum amount of information about featureclass layers is returned.

GET_SERVICE_INFO request setting attributes false:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_SERVICE_INFO envelope="false" fields="false" renderer="false"
extensions="false"/>
  </REQUEST>
</ARCXML>
```

When all the attributes are set to false in GET_SERVICE_INFO, the SERVICEINFO response returns the minimum amount of information for LAYERINFO. The ENVIRONMENT and PROPERTIES elements return the same information regardless of whether the attributes are true or false.

SERVICEINFO response with ENVELOPE:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
```

```
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="image" name="Background" visible="true" id="0">
        <ENVELOPE minx="-180" miny="-89.9747543334961"
maxx="179.9423828125" maxy="90" />
      </LAYERINFO>
      <LAYERINFO type="featureclass" visible="true" name="Countries"
id="1">
        <FCLASS type="polygon"></FCLASS>
      </LAYERINFO>
      <LAYERINFO type="acetate" name="northarrow" visible="true"
id="northarrow"/>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Envelope attribute

When envelope is set to true, the extent for each featureclass layer is included. This information is in addition to the extent of the service as a whole. The extent is always returned for image layers and no extent is included for acetate layers.

GET_SERVICE_INFO request using envelope:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_SERVICE_INFO envelope="true" fields="false" renderer="false"
extensions="false"/>
  </REQUEST>
</ARCXML>
```

In the SERVICEINFO response, an ENVELOPE is included for all featureclass layers. This is in addition to the ENVELOPE in the PROPERTIES section and the ENVELOPE for image layers. No ENVELOPE is included for acetate layers.

SERVICEINFO response with ENVELOPE:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
```

```
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="image" name="Background" visible="true" id="0">
        <ENVELOPE minx="-180" miny="-89.9747543334961"
maxx="179.9423828125" maxy="90" />
      </LAYERINFO>
      <LAYERINFO type="featureclass" visible="true" name="Countries"
id="1">
        <FCLASS type="polygon">
          <ENVELOPE minx="-180" miny="-90" maxx="180"
maxy="83.5960388183594" />
        </FCLASS>
      </LAYERINFO>
      <LAYERINFO type="acetate" name="northarrow" visible="true"
id="northarrow"/>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Fields attribute

When *fields* is set to "true" in GET_SERVICE_INFO, fields in the shapefile or ArcSDE table are returned in FIELD in the SERVICEINFO response. If joined tables are used, they are included in the FIELD list. In the request below, *fields* is set to "true". The response includes FIELD information for the Countries featureclass layer.

GET_SERVICE_INFO request with fields:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_SERVICE_INFO envelope="false" fields="true" renderer="false"
extensions="false"/>
  </REQUEST>
</ARCXML>
```

SERVICEINFO response with FIELD:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
```

```
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="image" name="Background" visible="true" id="0">
        <ENVELOPE minx="-180" miny="-89.9747543334961"
maxx="179.9423828125" maxy="90" />
      </LAYERINFO>
      <LAYERINFO type="featureclass" visible="true" name="Countries"
id="1">
        <FCLASS type="polygon">
          <FIELD name="AREA" type="8" size="12" precision="3" />
          <FIELD name="NAME" type="12" size="40" precision="0" />
          <FIELD name="ABBREVNAME" type="12" size="12" precision="0" />
          <FIELD name="FIPS_CODE" type="12" size="2" precision="0" />
          <FIELD name="WB_CNTRY" type="12" size="3" precision="0" />
          <FIELD name="#SHAPE#" type="-98" size="0" precision="0" />
          <FIELD name="#ID#" type="-99" size="16" precision="0" />
        </FCLASS>
      <LAYERINFO type="acetate" name="northarrow" visible="true"
id="northarrow"/>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Renderer attribute

When *renderer* is set to "true" in GET_SERVICE_INFO, all rendering information for a featureclass layer is included in the SERVICEINFO response. In the next example, *renderer* is set to "true". The response includes renderer information for the Countries featureclass layer. The returned information duplicates what is in the map configuration file.

GET_SERVICE_INFO request with renderer:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_SERVICE_INFO envelope="false" fields="false" renderer="true"
extensions="false"/>
  </REQUEST>
</ARCXML>
```

SERVICEINFO response with rendering:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
```

```
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="image" name="Background" visible="true" id="0">
        <ENVELOPE minx="-180" miny="-89.9747543334961"
maxx="179.9423828125" maxy="90" />
      </LAYERINFO>
      <LAYERINFO type="featureclass" visible="true" name="Countries"
id="1">
        <FCLASS type="polygon">
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL filltransparency="0.0"
boundarywidth="2" />
          </SIMPLERENDERER>
        </FCLASS>
      </LAYERINFO>
      <LAYERINFO type="acetate" name="northarrow" visible="true"
id="northarrow"/>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Extensions attribute

When *extensions* is set to "true" in GET_SERVICE_INFO, all extension information for a featureclass layer is included in the SERVICEINFO response. The three extension types are:

- **StoredQuery.** Defines data for a stored query using STOREDQUERY.
- **Extract.** Allows data to be extracted in shapefile format generated by the Extract Server. The extension is set up using EXTRACTPARAMS.
- **Geocode.** Identifies the geocoding style for a layer using GCSTYLE.

In the request below, *extensions* is set to "true". The response includes three extensions for the Countries featureclass layer. The returned information duplicates what is in the map configuration file for "StoredQuery" and "Extract". The response for the "Geocode" extension does not include everything in the map configuration file, only GCSTYLE.

GET_SERVICE_INFO request with extensions:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_SERVICE_INFO envelope="false" fields="false" renderer="false"
extensions="true"/>
  </REQUEST>
</ARCXML>
```

SERVICEINFO response with extensions:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="image" name="Background" visible="true" id="0">
        <ENVELOPE minx="-180" miny="-89.9747543334961"
maxx="179.9423828125" maxy="90" />
      </LAYERINFO>
      <LAYERINFO type="featureclass" visible="true" name="Countries"
id="1">
        <FCLASS type="polygon">
        <EXTENSION type="extract">
          <EXTRACTPARAMS clip="true">
            <OUTPUTFILE file="world">
              <OUTPUTFIELD name="NAME" alias="Country"/>
            </OUTPUTFILE>
          </EXTRACTPARAMS>
        </EXTENSION>
        <EXTENSION type="Geocode" >
          <GCSTYLE name="SingleField" />
        </EXTENSION>
        <EXTENSION type="StoredQuery">
          <STOREDQUERIES>
            <STOREDQUERY name="Country">
              <QUERY subfields="#SHAPE# AREA NAME ABBREVNAME FIPS_CODE
WB_CNTRY" where="( NAME = [%var%] )" >
              </QUERY>
              <SQVAR name="[%var%]" position="0">
                <FIELD name="NAME" type="12" precision="0" size="40" />
              </SQVAR>
            </STOREDQUERY>
          </STOREDQUERIES>
        </EXTENSION>
      </LAYERINFO>
      <LAYERINFO type="acetate" name="northarrow" visible="true"
id="northarrow"/>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Extensions attribute when GET_SERVICE_INFO is routed to the Geocode Server

In order to retrieve all information about a Geocode extension, the GET_SERVICE_INFO request must be routed to the Geocode Server. This routing information is contained in the URL sent to the ArcIMS site such as in the following example (all one line):

http://myComputer.domain.com/servlet/com.esri.esrimap.Esrimap?ClientVersion=3.1
&ServiceName=myservice
&**CustomService=Geocode**
&Form=True&Encode=True

The GET_SERVICE_INFO request is the same as in the previous example. However, by routing the request to the Geocode Server, complete geocode extension information is returned. This information is similar to the information in the map configuration file. However, where GCFIELD is used in the map configuration file, GCINPUT is used in the SERVICEINFO response.

GET_SERVICE_INFO request with extensions routed to Geocode Server:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_SERVICE_INFO envelope="false" fields="false" renderer="false"
extensions="true"/>
  </REQUEST>
</ARCXML>
```

SERVICEINFO response with Geocode extension information:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
      </ENVIRONMENT>
      <LAYERINFO name="Countries" id="1" >
        <EXTENSION type="geocode">
          <GCSTYLE name="SingleField" >
            <GCINPUT id="KEYFIELD" type="text" label="KeyField"
width="10" description="Any single field" />
          </GCSTYLE>
        </EXTENSION>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Dpi attribute

One final attribute in GET_SERVICE_INFO is *dpi* (dots per inch). If a value is assigned to *dpi* in GET_SERVICE_INFO, this new value is returned in SCREEN in the SERVICEINFO response. Based on the new dpi, the minimum and maximum scales returned for scale-dependent elements, such as SCALEDEPENDENTRENDERER, LAYERINFO, LAYER, and OBJECT, are recalculated. The recalculation is useful in situations where the map configuration file was created on a machine with one dpi, but the client uses a different dpi.

If *dpi* is not included in GET_SERVICE_INFO, the *dpi* set in SCREEN in the map configuration file is used. If SCREEN is not included in the map configuration file, a default value of "96" is assumed.

The ArcIMS Java Viewers, ArcExplorer 4, and ArcMap send the *dpi* of the client in a GET_SERVICE_INFO request. The ArcIMS HTML Viewer does not send the *dpi* of the client. It uses the SCREEN *dpi* in the map configuration file or a default value of "96". Therefore, when using the ArcIMS HTML Viewer, the dpi of the client may not have the same dpi as the map configuration file. This means that scale dependencies will behave slightly different in the HTML Viewer.

The minimum and maximum scale values in LAYERINFO are listed as number of map units per pixel. In requests and the map configuration file, scales can be set using a relative scale ratio such as 1:24000. In this example, one meter equals 24000 meters or one inch equals 24000 inches.

Map units per pixel refers to the number of meters, feet, or decimal degrees represented by one pixel in a map. To convert from a relative scale to map units per pixel, the size of a pixel must first be calculated. The formula for finding the number of meters in a pixel is 0.0254 / dpi where:

- The value 0.0254 is the number of meters in an inch.
- Dpi is the dpi set in the ArcIMS service or request. If no dpi is set in the service or request, the dpi is assumed to be 96.

As an example of calculating pixel size, if the dpi is 96, the pixel size is 0.0254 / 96, or 0.000265 m.

Once the pixel size is known, the relative scale can be converted to map units per pixel:

1. If the scale is in **meters**. To calculate the number of meters per pixel, take the relative scale and multiply by the pixel size (0.000265). For example, if the relative scale is 1:24000, then the number of meters per pixel is 24000 * 0.000265, or 6.36 meters.

2. If the scale is in **feet**. Do the calculation for meters (#1). Multiply the result by 3.28 (the number of feet in a meter). For example, if the number of meters per pixel is 6.36, the number of feet per pixel is 6.36 * 3.28, or 20.86 feet.
3. If the scale is in **decimal degrees**. For these calculations, the earth is assumed to be an exact circle with a circumference of 40030.174 km. One degree is 111.195 km (40030.174/360 degrees) or 111195 meters. To calculate the number of degrees, first do the calculation for meters (#1). Next, divide the result by 111195. For example, if the number of meters per pixel is 6.36, the number of degrees per pixel is 6.36 / 111195, or 0.0000571968.

If the *maxscale* for a LAYER is set to 1:12500000 in the map configuration file, and the *dpi* in GET_SERVICE_INFO is "96", then the *maxscale* for LAYERINFO is "3307.29828126323", or approximately 3307 meters per pixel.

```
<LAYERINFO type="featureclass" visible="true" name="Cities" id="4"
maxscale="3307.29828126323">
  <FCLASS type="point"> </FCLASS>
</LAYERINFO>
```

If the *dpi* in GET_SERVICE_INFO is "120", then the *maxscale* for LAYERINFO is "2645.83862501058", or approximately 2645 meters per pixel.

```
<LAYERINFO type="featureclass" visible="true" name="Cities" id="4"
maxscale="2645.83862501058">
  <FCLASS type="point"> </FCLASS>
</LAYERINFO>
```

The next examples show a GET_SERVICE_INFO request that includes *dpi* and the response.

GET_SERVICE_INFO request with dpi:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_SERVICE_INFO dpi="192" envelope="false" fields="false"
renderer="false" extensions="false"/>
  </REQUEST>
</ARCXML>
```

SERVICEINFO response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
```

```
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="192"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="image" name="Background" visible="true" id="0">
        <ENVELOPE minx="-180" miny="-89.9747543334961"
maxx="179.9423828125" maxy="90" />
      </LAYERINFO>
      <LAYERINFO type="featureclass" visible="true" name="Countries"
id="1">
        <FCLASS type="polygon"><FCLASS>
      </LAYERINFO>
      <LAYERINFO type="acetate" name="northarrow" visible="true"
id="northarrow"/>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Using GET_SERVICE_INFO and SERVICEINFO with ArcMap Image Services

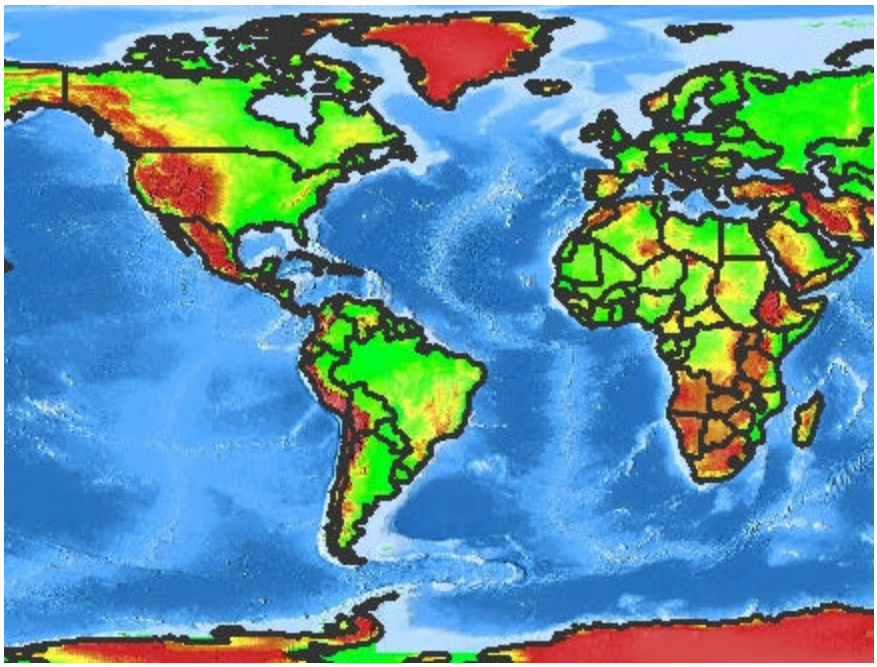
Introduction

GET_SERVICE_INFO is a request for accessing information about an ArcMap Image Service such as the environment, properties, layer, and dataframe information. This information is returned in a SERVICEINFO response. This document covers ArcMap Image Services. For information regarding Feature and Image Services, see Using GET_SERVICE_INFO and SERVICEINFO with Image Services.

ArcMap Image Services are created using ArcMap. The map document generated in ArcMap is the input to an ArcMap Image Service. All information about layer rendering is contained in the map document and is not included in the SERVICEINFO response. Also, because no information about geocode and extract extensions are included in a map document, GET_GEOCODE and GET_EXTRACT requests cannot be made to an ArcMap Image Service, and no extension information is included in a SERVICEINFO response. The following information from an ArcMap map document is used for the GET_SERVICE_INFO examples in this document. The document consists of two layers from the ESRIDATA data set. The following table summarizes the layer names, file name, file type, and layer ID number.

Layer Name	Shapefile Name	Data Type	Layer ID
Countries	CNTRY94	Polygon	0
Background	WORLD_IMG.gif	Image	1

The Background layer is an image of the world. The Countries layer contains the country boundaries. The map in the following figure is example output using this service.



GET_SERVICE_INFO Framework

A GET_SERVICE_INFO request requires only the GET_SERVICE_INFO element as shown in the example below.

GET_SERVICE_INFO request:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_SERVICE_INFO envelope="true" fields="true" />
  </REQUEST>
</ARCXML>
```

SERVICEINFO Framework

Depending on which attributes are used with GET_SERVICE_INFO, all or a subset of information about the ArcMap Image Service is returned in SERVICEINFO. Based on the above request on the sample service, the following example shows the SERVICEINFO response.

SERVICEINFO response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US"/>
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular"/>
```

```
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576"/>
        <CAPABILITIES forbidden=""/>
      </ENVIRONMENT>
      <LAYOUTINFO pageunits="inches">
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
      </LAYOUTINFO>
      <PROPERTIES>
        <FEATURECOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
        <FILTERCOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
        <MAPUNITS units="decimal_degrees"/>
        <BACKGROUND color="255,255,255"/>
        <ENVELOPE minx="-178.637433658708" miny="-149.932588181915"
maxx="180.202305702247" maxy="148.932531819555" name="Initial_Extent"
/>
      </PROPERTIES>
      <LAYERINFO type="image" name="Background" id="1" visible="true">
        <ENVELOPE minx="-180.10415" miny="-89.8896767396583"
maxx="179.900453479317" maxy="90.112625" />
      </LAYERINFO>
      <LAYERINFO type="featureclass" name="Countries" id="0"
visible="true">
        <FCLASS type="polygon">
          <ENVELOPE minx="-180" miny="-90" maxx="180"
maxy="83.5960388183594" />
          <FIELD name="#ID#" type="-99" size="4" precision="0" />
          <FIELD name="#SHAPE#" type="-98" size="0" precision="0" />
          <FIELD name="AREA" type="8" size="12" precision="11" />
          <FIELD name="NAME" type="12" size="40" precision="0" />
          <FIELD name="ABBREVNAME" type="12" size="12" precision="0" />
          <FIELD name="FIPS_CODE" type="12" size="2" precision="0" />
          <FIELD name="WB_CNTRY" type="12" size="3" precision="0" />
        </FCLASS>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

This SERVICEINFO response is divided into four sections: ENVIRONMENT, LAYOUTINFO, PROPERTIES, and LAYERINFO. A fifth child element, DATAFRAMEINFO, is discussed in a later section.

ENVIRONMENT

ENVIRONMENT takes its values from the ArcMap Image Service or assigns default values. LOCALE and UIFONT are required, so the information in SERVICEINFO is always taken directly from the service. With SCREEN and SEPARATORS, the values are either those assigned in the service or default values assigned by the ArcIMS Spatial Server. For SCREEN, the default value for *dpi* is 96. For SEPARATORS, the default for *cs* (coordinate separator) and *ts* (tuple separator) are a space and semicolon, respectively.

IMAGELIMIT is always included and is set when an ArcMap Image Service is started. By default, an image is limited in size to 4 MB, which corresponds to 1,048,576 pixels. Changes can be made to the image size and pixel count using the ArcIMS Administrator. For more information, see *Using ArcIMS*. The attribute *pixelcount* reports the pixel count from the ArcIMS Administrator, but it cannot be changed using GET_SERVICE_INFO.

CAPABILITIES contains a listing of which forbidden elements are added to an access control list (ACL). The ACL file is used for user authentication to grant users access to specific ArcMap Image Services. One way to limit access is to forbid one or more of the requests: GET_FEATURES, GET_IMAGE, GET_SERVICE_INFO, GET_LAYOUT, or GET_RASTER_INFO,. For more information on forbidden elements, see the ArcIMS Help.

In the example below, *forbidden* in CAPABILITIES is set to GET_LAYOUT. This means no requests with GET_LAYOUT are processed.

ENVIRONMENT and its child elements:

```
<ENVIRONMENT>
  <LOCALE language="en" country="US" />
  <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
  <SEPARATORS cs=" " ts=";"/>
  <SCREEN dpi="96"/>
  <IMAGELIMIT pixelcount="1048576" />
  <CAPABILITIES forbidden="GET_LAYOUT"/>
</ENVIRONMENT>
```

LAYOUTINFO

LAYOUTINFO in SERVICEINFO provides the initial envelope and units for an ArcMap Image Service layout. The coordinates for ENVELOPE are in page units of the layout rather than map units. Note that this ENVELOPE is the first one in the SERVICEINFO response. Prior to ArcIMS 4, the first ENVELOPE was in the PROPERTIES section. Its coordinates were in map units, and it contained the initial extent of the service. When determining the initial extent, be sure you are parsing the correct ENVELOPE - the ENVELOPE in the PROPERTIES section.

LAYOUTINFO and its child elements:

```
<LAYOUTINFO pageunits="inches">
  <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
</LAYOUTINFO>
```

PROPERTIES

The PROPERTIES section in SERVICEINFO returns properties of an ArcMap Image Service. The ENVELOPE is the initial extent for the service and is always included. All other child elements are optional in a service and are included in the SERVICEINFO response only when included in the service. Optional child elements include FEATURECOORDSYS, FILTERCOORDSYS, MAPUNITS, and BACKGROUND.

PROPERTIES and its child elements:

```
<PROPERTIES>
  <FEATURECOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
  <FILTERCOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
  <MAPUNITS units="decimal_degrees"/>
  <BACKGROUND color="255,255,255"/>
  <ENVELOPE minx="-178.637433658708" miny="-149.932588181915"
maxx="180.202305702247" maxy="148.932531819555" name="Initial_Extent"
/>
</PROPERTIES>
```

LAYERINFO

LAYERINFO describes the layer information in a service and includes the layer type, name of the layer, whether the layer is visible, the layer ID, and the minimum and maximum scales at which the layer can be displayed. All these parameters are defined in the ArcMap map document.

There are two LAYERINFO types:

- **Image.** In addition to LAYERINFO, the child element ENVELOPE is always included with an image layer.
- **Featureclass.** In addition to LAYERINFO, featureclass layers always include the child element FCLASS, which describes whether the featureclass is a polygon, line, or point. Also, as discussed in the next section, depending on different

attributes selected in GET_SERVICE_INFO, additional information about the featureclass layer can be included in the SERVICEINFO response.

The following example shows each of the layer types with the minimum required information.

LAYERINFO and minimum returned information:

```
<LAYERINFO type="image" name="Background" id="1" visible="true">
</LAYERINFO>
<LAYERINFO type="featureclass" name="Countries" id="0" visible="true">
  <FCLASS type="polygon"></FCLASS>
</LAYERINFO>
```

GET_SERVICE_INFO Attributes

GET_SERVICE_INFO has six optional attributes.

- envelope
- fields
- dataframe
- toc and toctype
- dpi

Envelope attribute

When envelope is set to "true", the extent for each featureclass layer is included in the SERVICEINFO response.

GET_SERVICE_INFO request using envelope:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_SERVICE_INFO envelope="true" fields="false" />
  </REQUEST>
</ARCXML>
```

In the response, an ENVELOPE is included for each layer. This is in addition to the ENVELOPE in the PROPERTIES and LAYOUTINFO sections. Remember that the envelope in the LAYOUTINFO section is in page units. All other envelopes are in map units.

SERVICEINFO response with ENVELOPE:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
```

```
      <ENVIRONMENT>
        <LOCALE language="en" country="US"/>
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular"/>
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576"/>
        <CAPABILITIES forbidden=""/>
      </ENVIRONMENT>
      <LAYOUTINFO pageunits="inches">
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
      </LAYOUTINFO>
      <PROPERTIES>
        <FEATURECOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
        <FILTERCOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
        <MAPUNITS units="decimal_degrees"/>
        <BACKGROUND color="255,255,255"/>
        <ENVELOPE minx="-178.637433658708" miny="-149.932588181915"
maxx="180.202305702247" maxy="148.932531819555" name="Initial_Extent"
/>
      </PROPERTIES>
      <LAYERINFO type="image" name="Background" id="1" visible="true">
        <ENVELOPE minx="-180.10415" miny="-89.8896767396583"
maxx="179.900453479317" maxy="90.112625" />
      </LAYERINFO>
      <LAYERINFO type="featureclass" name="Countries" id="0"
visible="true">
        <FCLASS type="polygon">
          <ENVELOPE minx="-180" miny="-90" maxx="180"
maxy="83.5960388183594" />
        </FCLASS>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Fields attribute

When *fields* is set to "true" in GET_SERVICE_INFO, fields for all featureclass layers are returned in FIELD in the SERVICEINFO response. If joined tables are used, they are included in the FIELD list. In the request below, *fields* is set to "true". The response includes FIELD information for the Countries featureclass layer.

GET_SERVICE_INFO request with fields:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
```

```
    <GET_SERVICE_INFO envelope="false" fields="true" />
  </REQUEST>
</ARCXML>
```

SERVICEINFO response with FIELD:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US"/>
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular"/>
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576"/>
        <CAPABILITIES forbidden=""/>
      </ENVIRONMENT>
      <LAYOUTINFO pageunits="inches">
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
      </LAYOUTINFO>
      <PROPERTIES>
        <FEATURECOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
        <FILTERCOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
        <MAPUNITS units="decimal_degrees"/>
        <BACKGROUND color="255,255,255"/>
        <ENVELOPE minx="-178.637433658708" miny="-149.932588181915"
maxx="180.202305702247" maxy="148.932531819555" name="Initial_Extent"
/>
      </PROPERTIES>
      <LAYERINFO type="image" name="Background" id="1" visible="true">
      </LAYERINFO>
      <LAYERINFO type="featureclass" name="Countries" id="0"
visible="true">
        <FCLASS type="polygon">
          <FIELD name="#ID#" type="-99" size="4" precision="0" />
          <FIELD name="#SHAPE#" type="-98" size="0" precision="0" />
          <FIELD name="AREA" type="8" size="12" precision="11" />
          <FIELD name="NAME" type="12" size="40" precision="0" />
          <FIELD name="ABBREVNAME" type="12" size="12" precision="0" />
          <FIELD name="FIPS_CODE" type="12" size="2" precision="0" />
          <FIELD name="WB_CNTRY" type="12" size="3" precision="0" />
        </FCLASS>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Dataframe attribute and DATAFRAMEINFO

ArcMap documents can contain multiple data frames. By default, only information on the active data frame is included in a SERVICEINFO response. To retrieve information on all data frames, the *dataframe* attribute must be used. When used, the structure of the response changes, and DATAFRAMEINFO is included in the response. The following table compares the two structures in general terms.

Request	Structure

When *dataframe* is not included.	`<?xml version="1.0" encoding="UTF-8"?>`
	`<ARCXML version="1.1">`
	`  <RESPONSE>`
	`  <SERVICEINFO>`
	`  <ENVIRONMENT>...</ENVIRONMENT>`
	`  <LAYOUTINFO>...</LAYOUTINFO>`
	`  **<PROPERTIES>...</PROPERTIES>**`
	`  **<LAYERINFO>...</LAYERINFO>**`
	`  **<LAYERINFO>...</LAYERINFO>**`
	`  </SERVICEINFO>`
	`  </RESPONSE>`
	`</ARCXML>`

When *dataframe* is included.	`<?xml version="1.0" encoding="UTF-8"?>`
	`<ARCXML version="1.1">`
	`  <RESPONSE>`
	`  <SERVICEINFO>`
	`  <ENVIRONMENT>...</ENVIRONMENT>`
	`  <LAYOUTINFO>...</LAYOUTINFO>`
	`  **<DATAFRAMEINFO>**`
	`  **<PROPERTIES>...</PROPERTIES>**`
	`  **<LAYERINFO>...</LAYERINFO>**`
	`  **</DATAFRAMEINFO>**`
	`  **<DATAFRAMEINFO>**`
	`  **<PROPERTIES>...</PROPERTIES>**`
	`  **<LAYERINFO>...</LAYERINFO>**`
	`  **</DATAFRAMEINFO>**`
	`  </SERVICEINFO>`
	`  </RESPONSE>`
	`</ARCXML>`

Since each data frame has its own properties and layer information, when *dataframe* is included in the request, PROPERTIES and LAYERINFO become child elements to DATAFRAMEINFO in the response.

Dataframe takes data frame names as its value. The following table summarizes different scenarios for assigning values to *dataframe*:

Usage	Result

dataframe=""	Treated the same as if *dataframe* were not present in request.
dataframe="#ALL#"	All data frames are included in response.
dataframe="Layers"	Only information on the data frame named "Layers" is included in response.
dataframe="Layers;States"	Information on the data frames named "Layers" and "States" is included in response. The separator used is a semicolon (;). One restriction in the ArcMap document is that data frame names cannot include a semicolon. If a semicolon is included in the data frame name, the request will likely not process correctly.
dataframe="Layersxxx"	If a data frame in the list does not exist, an error message is returned, and the request is not processed.

In the next example, *dataframe* is set to "#ALL#".

GET_SERVICE_INFO request with renderer:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_SERVICE_INFO envelope="false" fields="false" dataframe="#ALL#"
/>
  </REQUEST>
</ARCXML>
```

Since only one data frame is included in the service, only one instance of DATAFRAMEINFO is included in the response.

SERVICEINFO response with DATAFRAMEINFO:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US"/>
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular"/>
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576"/>
        <CAPABILITIES forbidden=""/>
      </ENVIRONMENT>
      <LAYOUTINFO pageunits="inches">
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
```

```
      </LAYOUTINFO>
      <DATAFRAMEINFO name="Layers">
        <PROPERTIES>
          <FEATURECOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
          <FILTERCOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
          <MAPUNITS units="decimal_degrees"/>
          <BACKGROUND color="255,255,255"/>
          <ENVELOPE minx="-176.453309664892" miny="-120.899619477762"
maxx="174.852769885559" maxy="123.217832626079" name="Initial_Extent"
/>
        </PROPERTIES>
        <LAYERINFO type="image" name="Background" id="1"
visible="true">
          <ENVELOPE minx="-180.10415" miny="-89.8896767396583"
maxx="179.900453479317" maxy="90.112625" />
        </LAYERINFO>
        <LAYERINFO type="featureclass" name="Countries" id="0"
visible="true">
          <FCLASS type="polygon">
            <ENVELOPE minx="-180" miny="-90" maxx="180"
maxy="83.5960388183594" />
            <FIELD name="FID" type="-99" size="4" precision="0" />
            <FIELD name="#SHAPE#" type="-98" size="0" precision="0" />
            <FIELD name="AREA" type="8" size="12" precision="11" />
            <FIELD name="NAME" type="12" size="40" precision="0" />
            <FIELD name="ABBREVNAME" type="12" size="12" precision="0"
/>
            <FIELD name="FIPS_CODE" type="12" size="2" precision="0" />
            <FIELD name="WB_CNTRY" type="12" size="3" precision="0" />
          </FCLASS>
        </LAYERINFO>
      </DATAFRAMEINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Toc and toctype attributes

When *toc* is set to "true" in GET_SERVICE_INFO, information for swatches used in the table of contents for ArcIMS Java Viewers, ArcExplorer 4, and ArcMap is included. *Toctype* specifies the format of the swatches. In the request below, *toc* is set to true and *toctype* is set to "bmp".

GET_SERVICE_INFO request with extensions:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_SERVICE_INFO envelope="false" fields="false" toc="true"
toctype="bmp"/>
  </REQUEST>
</ARCXML>
```

The response includes TOC, TOCGROUP, and TOCCLASS. Together, these elements define the swatches and any text used with the swatches. The swatch images embedded in the TOCCLASS are base64 encoded and are not compressed. The response can be rather lengthy if many layers are used and many categories are used within each layer. The following response shows the TOCCLASS information for the two layers in the service. Note that most of the embedded information has been removed.

SERVICEINFO response with extensions:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US"/>
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular"/>
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576"/>
        <CAPABILITIES forbidden=""/>
      </ENVIRONMENT>
      <LAYOUTINFO pageunits="inches">
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
      </LAYOUTINFO>
      <PROPERTIES>
        <FEATURECOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
        <FILTERCOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
        <MAPUNITS units="decimal_degrees"/>
        <BACKGROUND color="255,255,255"/>
        <ENVELOPE minx="-178.637433658708" miny="-149.932588181915"
maxx="180.202305702247" maxy="148.932531819555" name="Initial_Extent"
/>
      </PROPERTIES>
      <LAYERINFO type="image" name="Background" id="1" visible="true">
        <TOC>
          <TOCGROUP heading="RGB Composite">
            <TOCCLASS label="Red: Band_1" description="Red:
Band_1"<Qk0SCQ ...5uAAAA</TOCCLASS>
```

```
            <TOCCLASS label="Red: Band_2" description="Red:
Band_2"<Qk0SCQ ...5uAAAA</TOCCLASS>
            <TOCCLASS label="Red: Band_3" description="Red:
Band_3"<Qk0SCQ ...5uAAAA</TOCCLASS>
          </TOCGROUP>
        </TOC>
      </LAYERINFO>
      <LAYERINFO type="featureclass" name="Countries" id="0"
visible="true">
        <FCLASS type="polygon"></FCLASS>
        <TOC>
          <TOCGROUP>
            <TOCCLASS label="" description="">Qk0SCQ
...5uAAAA</TOCCLASS>
          </TOCGROUP>
        </TOC>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Dpi attribute

The final attribute in GET_SERVICE_INFO is *dpi* (dots per inch). If a value is assigned to *dpi* in GET_SERVICE_INFO, this new value is returned in SCREEN in the SERVICEINFO response. Based on the new dpi, the minimum and maximum scales returned for LAYERINFO are recalculated. The recalculation is useful in situations where the ArcMap document was created on a machine with one dpi, but the client uses a different dpi.

If *dpi* is not included in GET_SERVICE_INFO, the *dpi* in the ArcMap document is used. If SCREEN is not included in the map configuration file, a default value of "96" is assumed.

The ArcIMS Java Viewers, ArcExplorer 4, and ArcMap send the *dpi* of the client in a GET_SERVICE_INFO request. The ArcIMS HTML Viewer does not send the *dpi* of the client. It uses the SCREEN *dpi* in the map configuration file or a default value of "96". Therefore, when using the ArcIMS HTML Viewer, the dpi of the client may not have the same dpi as the map configuration file. This means that scale dependencies will behave slightly different in the HTML Viewer.

The minimum and maximum scale values in LAYERINFO are listed as number of map units per pixel. In requests and the map configuration file, scales can be set using a relative scale ratio such as 1:24000. In this example, 1 meter equals 24000 meters or 1 inch equals 24000 inches.

Map units per pixel refers to the number of meters, feet, or decimal degrees represented by one pixel in a map. To convert from a relative scale to map units per pixel, the size of a pixel must first be calculated. The formula for finding the number of meters in a pixel is 0.0254 / dpi where:

- The value 0.0254 is the number of meters in an inch.
- Dpi is the dpi set in the ArcIMS service or request. If no dpi is set in the service or request, the dpi is assumed to be 96.

As an example of calculating pixel size, if the dpi is 96, the pixel size is 0.0254 / 96 or 0.000265 m.

Once the pixel size is known, the relative scale can be converted to map units per pixel:

1. If the scale is in **meters**. To calculate the number of meters per pixel, take the relative scale and multiply by the pixel size (0.000265). For example, if the relative scale is 1:24000, then the number of meters per pixel is 24000 * 0.000265, or 6.36 meters.
2. If the scale is in **feet**. Do the calculation for meters (#1). Multiply the result by 3.28 (the number of feet in a meter). For example, if the number of meters per pixel is 6.36, the number of feet per pixel is 6.36 * 3.28, or 20.86 feet.
3. If the scale is in **decimal degrees**. For these calculations, the earth is assumed to be an exact circle with a circumference of 40030.174 km. One degree is 111.195 km (40030.174/360 degrees) or 111195 meters. To calculate the number of degrees, first do the calculation for meters (#1). Next, divide the result by 111195. For example, if the number of meters per pixel is 6.36, the number of degrees per pixel is 6.36 / 111195, or 0.0000571968.

If the *maxscale* for a LAYER is set to 1:12500000 in the map configuration file, and the *dpi* in GET_SERVICE_INFO is "96", then the *maxscale* for LAYERINFO is "3307.29828126323", or approximately 3307 meters per pixel.

```
<LAYERINFO type="featureclass" name="Countries" id="0" visible="true"
maxscale="3307.29828126323">
  <FCLASS type="point"> </FCLASS>>
</LAYERINFO>
```

If the *dpi* in GET_SERVICE_INFO is "120", then the *maxscale* for LAYERINFO is "2645.83862501058", or approximately 2645 meters per pixel.

```
<LAYERINFO type="featureclass" name="Countries" id="0" visible="true"
maxscale="2645.83862501058">
  <FCLASS type="point"> </FCLASS>>
</LAYERINFO>
```

The next examples show a GET_SERVICE_INFO request that includes dpi.

GET_SERVICE_INFO request with dpi:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
```

```
  <REQUEST>
    <GET_SERVICE_INFO dpi="192" envelope="false" fields="false" />
  </REQUEST>
</ARCXML>
```

The response includes the dpi in SCREEN, and recalculated scale threshold values in LAYERINFO. In this example, the service includes three layers and is in decimal degrees.

SERVICEINFO response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US"/>
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular"/>
        <SEPARATORS cs=" " ts=";"/>
        <IMAGELIMIT pixelcount="1048576"/>
        <SCREEN dpi="192"/>
        <CAPABILITIES forbidden=""/>
      </ENVIRONMENT>
      <LAYOUTINFO pageunits="inches">
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
      </LAYOUTINFO>
      <PROPERTIES>
        <FEATURECOORDSYS
string="GEOGCS[&quot;GCS_North_American_1983&quot;,DATUM[&quot;D_North_
American_1983&quot;,SPHEROID[&quot;GRS_1980&quot;,6378137.0,298.2572221
01]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.017453
2925199433]]"/>
        <FILTERCOORDSYS
string="GEOGCS[&quot;GCS_North_American_1983&quot;,DATUM[&quot;D_North_
American_1983&quot;,SPHEROID[&quot;GRS_1980&quot;,6378137.0,298.2572221
01]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.017453
2925199433]]"/>
        <MAPUNITS units="decimal_degrees"/>
        <BACKGROUND color="255,255,255"/>
        <ENVELOPE minx="-183.780014728684" miny="0.589718314252654"
maxx="-61.4068546696843" maxy="89.741298838335" name="Initial_Extent"
/>
      </PROPERTIES>
      <LAYERINFO type="featureclass" name="STATES" id="2"
visible="true">
        <FCLASS type="polygon"></FCLASS>
      </LAYERINFO>
      <LAYERINFO type="featureclass" name="COUNTIES" id="1"
visible="true" maxscale="0.011897304807224">
        <FCLASS type="polygon"></FCLASS>
      </LAYERINFO>
      <LAYERINFO type="featureclass" name="CITIES" id="0"
visible="true" minscale="0.002379460961445"
maxscale="0.014276765768669">
```

```
        <FCLASS type="point"></FCLASS>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Using GET_IMAGE and IMAGE with Image Services

Introduction

This document covers GET_IMAGE and IMAGE when using Image Services. When using ArcMap Image Services, see Using GET_IMAGE and GET_LAYOUT with ArcMap Image Services.

The purpose of GET_IMAGE and IMAGE is to render a map image on the ArcIMS Spatial Server and provide the location and filename of that image. GET_IMAGE is for generating a map only. To retrieve attribute data associated with the map, a separate GET_FEATURES request must be made.

GET_IMAGE requests are sent to Image Services. The service provides a default view of the data. Using GET_IMAGE, the service can be manipulated. Information such as map extent or layer rendering can be modified by the request. A service can be modified in one of two ways:

- By overriding the Image Service defaults by adding new parameters in the PROPERTIES section.
- By adding dynamic layers to the Image Service using LAYER.

Most of the sample requests are based on the following map configuration file. The file consists of five layers from the ESRIDATA dataset. The following table summarizes the layer names, shapefile name, and layer ID number.

Layer Name	Shapefile Name	Layer ID
Ocean	WORLD30	0
Countries	CNTRY94	1
States	STATES	2
Provinces	PROVINCE	3
Cities	CITIES	4

Image Service used with most of the GET_IMAGE requests that follow:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP dynamic="true" >
      <PROPERTIES>
```

```
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="C:\ESRIDATA\WORLD"
/>
        <SHAPEWORKSPACE name="shp_ws-2" directory="C:\ESRIDATA\USA" />
        <SHAPEWORKSPACE name="shp_ws-3" directory="C:\ESRIDATA\CANADA"
/>
      </WORKSPACES>
      <LAYER type="featureclass" name="Ocean" visible="true" id="0">
        <DATASET name="WORLD30" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="0,153,255"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Countries" visible="true"
id="1">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid"
fillcolor="255,255,153"/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="States" visible="true" id="2">
        <DATASET name="STATES" type="polygon" workspace="shp_ws-2" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Provinces" visible="true"
id="3">
        <DATASET name="province" type="polygon" workspace="shp_ws-3" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="0,153,0" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Cities" visible="true" id="4">
        <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="102,0,102" width="8.0" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

GET_IMAGE Request and IMAGE Response

The simplest GET_IMAGE request includes PROPERTIES with no child elements inside. With this request, a map image is generated using default rendering and extents established in the Image Service.

Simple GET_IMAGE request:

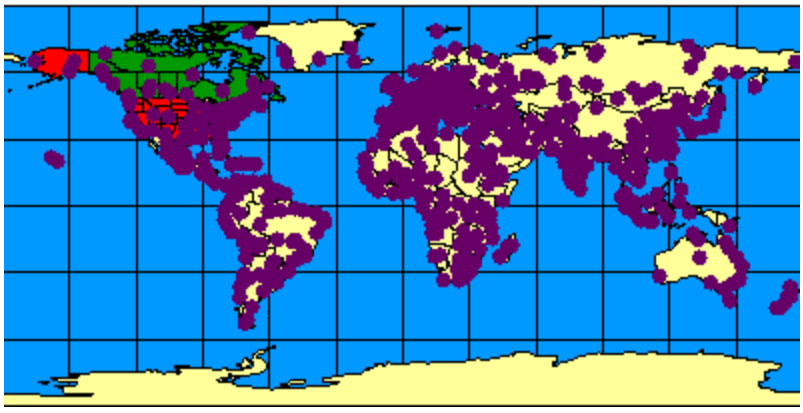
```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The IMAGE response includes a default envelope and the name and location of the generated map image.

IMAGE response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-180" miny="-144" maxx="180" maxy="144" />
      <OUTPUT file="c:\arcims\output\world_MYCOMPUTER2102209.png"
url="http://mycomputer.domain.com/output/world_MYCOMPUTER2102209.png"
/>
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

The returned map image is generated using defaults from the Image Service. The visible layers shown are Ocean, Countries, States, Provinces, and Cities. Also, the map is in geographic coordinates, and the default envelope includes the entire world.



Using show

When a GET_IMAGE request is made, additional information about a layer, such as the layer name, ID, and number of features sent, can be included in the response by setting the attribute *show*. In the next example, the attribute *show="layers"* is included with GET_IMAGE.

GET_IMAGE request with show="layers":

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE show="layers">
      <PROPERTIES>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The IMAGE response includes the layer information in addition to the name and location of the map image. The *featurecount* attribute includes only the number of features returned in the image.

IMAGE response with layer information:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-180" miny="-135" maxx="180" maxy="135" />
      <LAYERS>
        <LAYER name="Ocean" id="0" featurecount="72" />
        <LAYER name="Countries" id="1" featurecount="165" />
        <LAYER name="States" id="2" featurecount="51" />
        <LAYER name="Provinces" id="3" featurecount="12" />
        <LAYER name="Cities" id="4" featurecount="606" />
      </LAYERS>
      <OUTPUT file="c:\arcims\output\world_MYCOMPUTER3633699.jpg"
url="http://mycomputer.domain.com/output/world_MYCOMPUTER3633699.jpg"
/>
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

Using autoresize

During a GET_IMAGE request, the maximum size of an image can be no greater than the image memory limit set when an Image Service is started. For example, an image memory limit of 1 MB allows a map no larger than 262,144 pixels (512 x 512) to be generated. By default, an image is limited in size to 4 MB, which corresponds to 1,048,576 pixels. Changes can be made to the image size and pixel count using the

ArcIMS Administrator. For more information, see *Using ArcIMS*.

If *autoresize* is set to "true" in GET_IMAGE, a requested map greater than the maximum pixel count is reduced in size to within the maximum pixel count. In the next example, the Image Service image limit is 4 MB. IMAGESIZE requests an image greater than 4 MB. (IMAGESIZE is discussed in greater detail in the Changing the Output IMAGESIZE Section.)

GET_IMAGE request with autoresize="true":

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE autoresize="true">
      <PROPERTIES>
        <IMAGESIZE width="2000" height="1600" >
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The IMAGE response includes a resized image because the requested image size was greater than allowed. In OUTPUT, the attributes *height* and *width* show the new image size. (OUTPUT is discussed in greater detail in the Using OUTPUT to Control Image Names and Locations Section.)

IMAGE response with resized image information:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-180" miny="-135" maxx="180" maxy="135" />
      <OUTPUT file="c:\arcims\output\world_MYCOMPUTER3633699.jpg"
url="http://mycomputer.domain.com/output/world_MYCOMPUTER3633699.jpg
width="1086" height="965" />
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

If *autoresize* is "false" or not included in the GET_IMAGE request, and the requested image is too big, an error message is returned.

IMAGE response when requested map is too large:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <ERROR machine="MYMACHINE" processid="1324"
threadid="1904">Requested image is too big and can not be
created</ERROR>
  </RESPONSE>
</ARCXML>
```

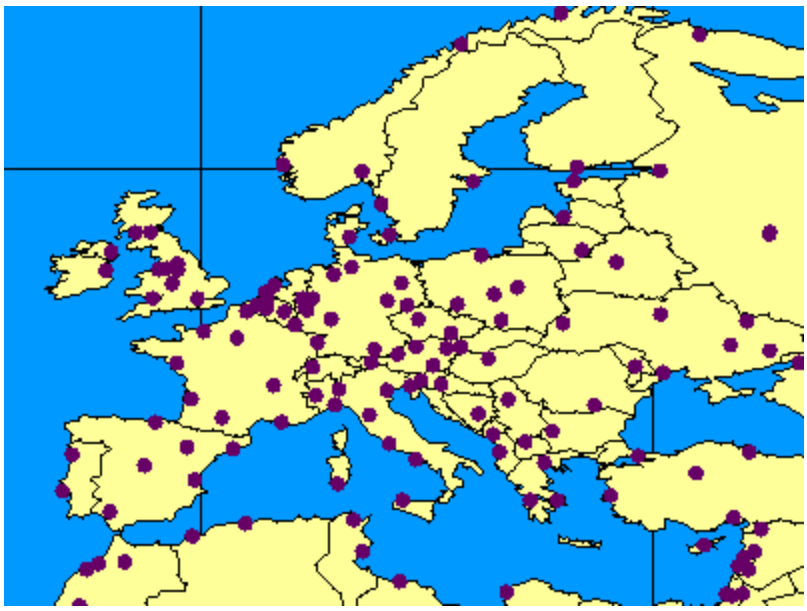
Zoom and Pan with ENVELOPE

When a user pans or zooms on a map, what really happens is that the map extent changes. In a GET_IMAGE request, the extent is changed by sending new x,y minimum and maximum coordinates in an ENVELOPE. This new envelope overrides the envelope set in the Image Service. In the following example, the envelope zooms into the region around Europe.

GET_IMAGE request with a change in ENVELOPE:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-13" miny="37" maxx="40" maxy="65" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The returned map image includes only Europe. All other information on the layers continues to use defaults from the Image Service.



Changing the Output IMAGESIZE

IMAGESIZE sets the size of the output map image in pixels. If IMAGESIZE is not used in a request, the default image size is 400 x 300 pixels. As noted in the using autoresize section, the maximum size of an image can be no greater than the image memory limit set when an Image Service is started.

The output image sizes can be controlled three ways using the following attribute groups:

- *width* and *height*
- *printwidth* and *printheight*
- *width*, *height*, and *dpi*

The following example, using a New York City street Image Service, shows a GET_IMAGE request with IMAGESIZE. The *width* and *height* are set to 250 and 175 pixels, respectively.

GET_IMAGE request using IMAGESIZE:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-73.985" miny="40.756" maxx="-73.972"
maxy="40.765" />
        <IMAGESIZE width="250" height="175"/>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Depending on the combination of IMAGESIZE attributes and ENVELOPE size used in the request, the results of the returned map image may differ.

Width and height

Width and *height* are required attributes. They are used to draw a map at the specified width and height based on the envelope. If *width* and *height* are changed but the envelope remains the same, the scale of the map changes. In other words, if *width* and *height* are increased, the map is larger and, in effect, "zoomed in". If a scale threshold is met, the symbology of a layer may change or a layer might be added or removed.

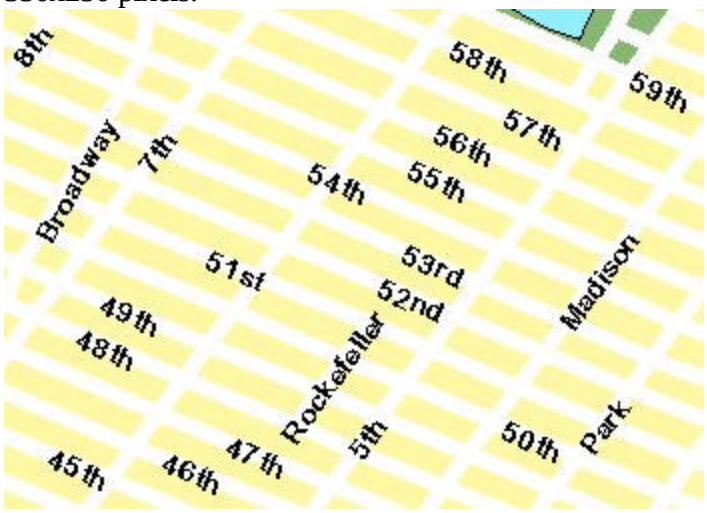
In the two images below, a set of streets in New York City is shown. The first map is 250 x 175 pixels in size. The second map has the same extent but is 350 x 250 pixels. Note that symbology has changed for the streets. By making the map larger and zooming in, a scale threshold was met instructing the ArcIMS Spatial Server to change the street symbols.

250x175 pixels:




```
<PROPERTIES>
  <ENVELOPE minx="-73.985"
miny="40.756" maxx="-
73.972" maxy="40.765" />
  <IMAGESIZE width="250"
height="175" />
</PROPERTIES>
```

350x250 pixels:




```
<PROPERTIES>
  <ENVELOPE minx="-73.985"
miny="40.756" maxx="-
73.972" maxy="40.765" />
  <IMAGESIZE width="350"
height="250" />
</PROPERTIES>
```

Printwidth and printheight

To avoid having the symbology change or having layers added or removed when the map size is increased or decreased, two sets of attributes are available: *printwidth* and *printheight* or *width*, *height*, and *dpi*. By using one of these attribute sets, the output image contains the same information regardless of the size.

In the examples below, *printwidth* and *printheight* are used. The scale factors and dependencies are calculated based on the *height* and *width* attributes, and the print attributes instruct the spatial server what size to draw the image. In the examples below, the first image has the *printwidth* and *printheight* set the same as the *width* and *height*. In the second image, the *width* and *height* remain the same, but the *printwidth* and *printheight* are increased. The two images contain the same content even though they are different sizes. The ratio for calculating the print map is *printwidth* / *width*. In this example the ratio is 350 / 250 or 1.4. The output map is 1.4 times the size of the original map.

The height and width of the second and third images is the same but the size of the

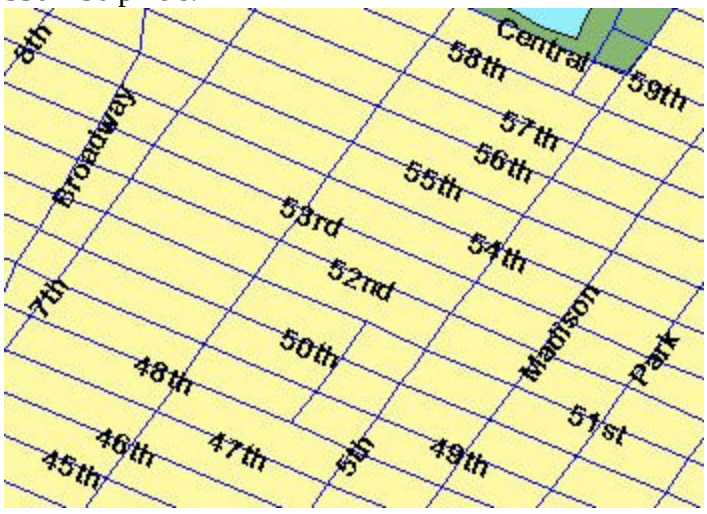
symbology is different. In order for symbols to scale, the attribute *scalesymbols* must be used. By default, the symbols do not scale in size when the map size is increased or decreased. The second image below shows the map with *scalesymbols* set to "false", and the labels are the same size as in the first image. The third image, on the other hand, has *scalesymbols* set to "true". When this attribute is used, the symbols increase and decrease in size proportionally as the image size increases or decreases. In the third example below, the symbology has increased proportionally in size so that it is also 1.4 times the size of the symbology in the original map.

250x175 pixels:



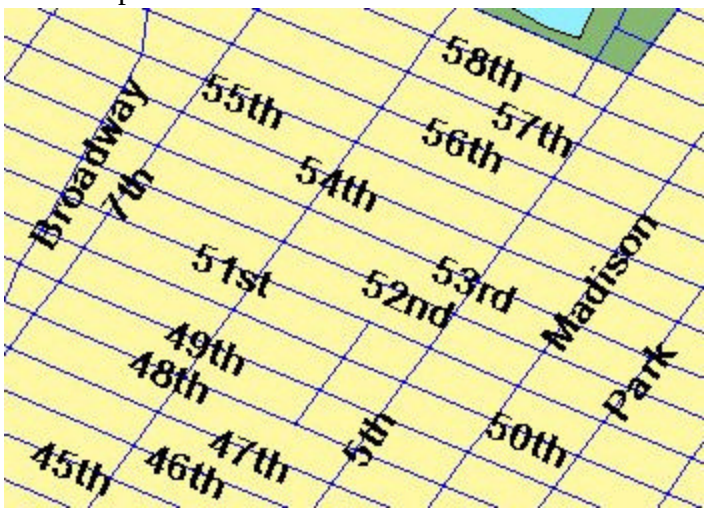

```
<PROPERTIES>
  <ENVELOPE minx="-73.985"
miny="40.756" maxx="-
73.972" maxy="40.765" />
  <IMAGESIZE width="250"
height="175"
printwidth="250"
printheight="175" />
</PROPERTIES>
```

350x250 pixels:




```
<PROPERTIES>
  <ENVELOPE minx="-73.985"
miny="40.756" maxx="-
73.972" maxy="40.765" />
  <IMAGESIZE width="250"
height="175"
printwidth="350"
printheight="250"
scalesymbols="false" />
</PROPERTIES>
```

350x250 pixels:




```
<PROPERTIES>
  <ENVELOPE minx="-73.985"
miny="40.756" maxx="-
73.972" maxy="40.765" />
  <IMAGESIZE width="250"
height="175"
printwidth="350"
printheight="250"
scalesymbols="true" />
</PROPERTIES>
```

Width, height, and dpi

The next examples use *width*, *height*, and *dpi* rather than *printwidth* and *printheight* but the final results are similar for calculating scale factors and dependencies. The first image is for reference and includes only *height* and *width*. In the second and third images, the *width* and *height* values are changed to the desired size of the output image, in this case 350 and 245 pixels, respectively. The *dpi* is set to "134". The ratio for calculating the output map is (*dpi* of the request) / (*dpi* of the Image Service). If *dpi* is not in the service, a default value of "96" is used. In this scenario, the *dpi* of the request is "134", and the *dpi* of the service is assumed to be "96". The ratio between the two is 134 / 96, or 1.4. This is the same ratio as the *printwidth* and *printheight* examples above, and the output image sizes are very similar.

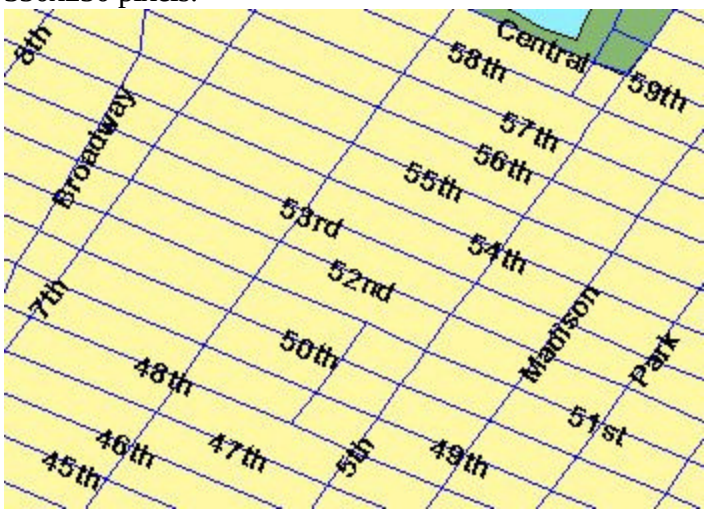
Once again, the symbology can be scaled as the map size increases or decreases by using the attribute *scalesymbols*. The second image below shows the map with *scalesymbols* set to "false" while in the third image *scalesymbols* set to "true".

250x175 pixels:



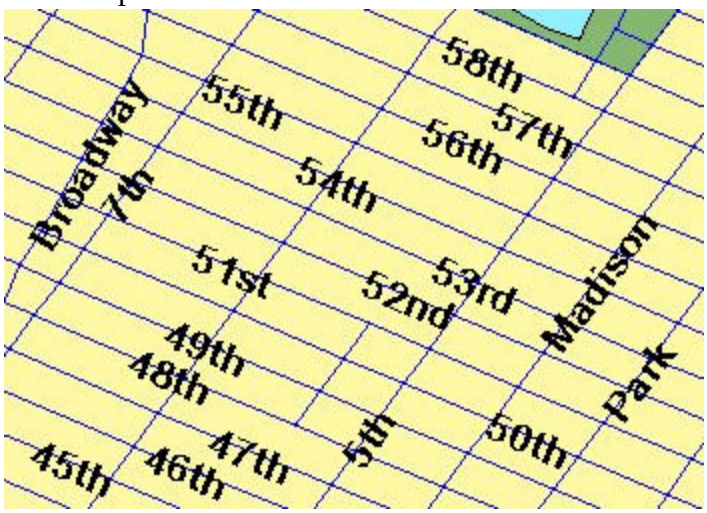

```
<PROPERTIES>
  <ENVELOPE minx="-73.985"
miny="40.756" maxx="-73.972"
maxy="40.765" />
  <IMAGESIZE width="250" height="175"
/>
</PROPERTIES>
```

350x250 pixels:




```
<PROPERTIES>
  <ENVELOPE minx="-73.985"
miny="40.756" maxx="-
73.972" maxy="40.765" />
  <IMAGESIZE width="350"
height="245" dpi="134"
scalesymbols="false" />
</PROPERTIES>
```

350x250 pixels:




```
<PROPERTIES>
  <ENVELOPE minx="-73.985"
miny="40.756" maxx="-
73.972" maxy="40.765" />
  <IMAGESIZE width="350"
height="245" dpi="134"
scalesymbols="true" />
</PROPERTIES>
```

Using BACKGROUND

BACKGROUND is used to define a color for the image background. It can also be used to make one color in the image transparent. Depending on the browser, the image formats that support transparent colors vary. JPG images do not support transparent colors. The table below lists which image formats support transparent colors for different browsers.

Browser	Supported Transparent Image Formats
ArcIMS HTML Viewer in Internet Explorer	PNG8, GIF
ArcIMS HTML Viewer in Netscape 4.x	GIF
ArcIMS HTML Viewer in Netscape 6.x	PNG8, GIF
ArcExplorer 4	PNG24, PNG8, GIF

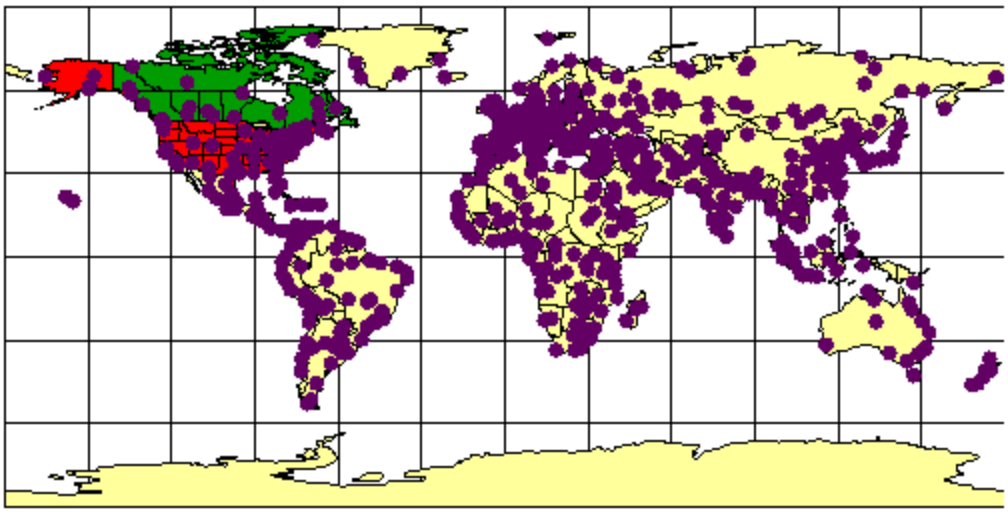
ArcIMS Java Viewers in Internet Explorer and Netscape	PNG24, PNG8, GIF

To make a color transparent, both the *color* and *transcolor* attributes of BACKGROUND must be set to the same color. In the following request, the transparent color is the blue color in the Ocean layer.

GET_IMAGE request using BACKGROUND:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <IMAGESIZE width="500" height="400" />
        <BACKGROUND color="0,153,255" transcolor="0,153,255" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

In the returned image, the ocean is now transparent.



LAYERLIST and LAYERDEF

LAYERLIST and LAYERDEF are used together in a GET_IMAGE request to manipulate how layers are drawn in the returned map image. Some common ways to manipulate the layers include:

- Setting layer visibility
- Changing layer symbology
- Querying a layer
- Changing layer order

Setting layer visibility

Layers in an Image Service can be switched on and off using LAYERDEF. If LAYERDEF is not included, the layer visibility is set to the visibility of the layers in the Image Service. Within the LAYERDEF element, layers that are set to *visible="false"* are not included in the image.

The layers in LAYERDEF are identified by their ID. The following table lists which ID corresponds to which layer.

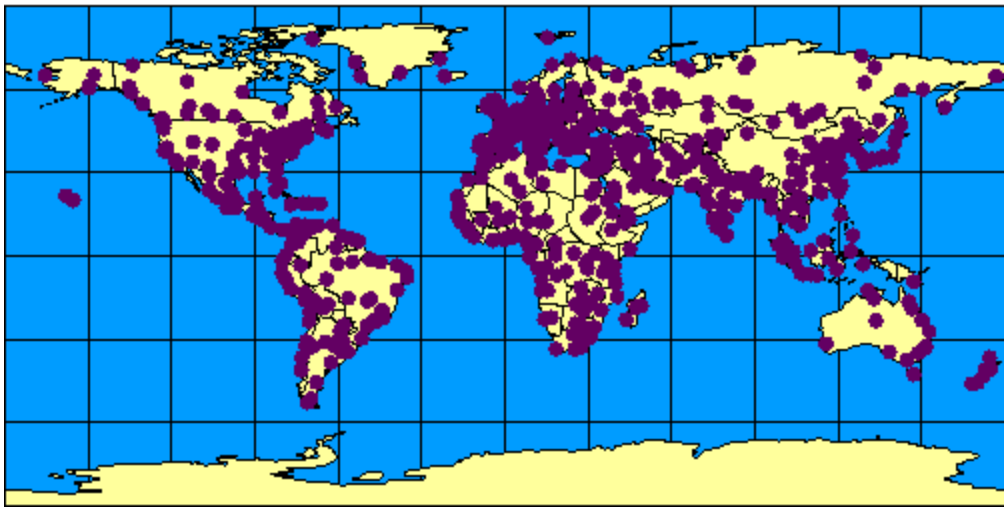
ID	Layer Name
0	Ocean
1	Countries
2	States
3	Provinces
4	Cities

In the following example, the States and Provinces layers have their visibility set to "false".

GET_IMAGE using LAYERDEF to set layer visibility:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <IMAGESIZE width="500" height="400" />
        <LAYERLIST>
          <LAYERDEF id="0" visible="true" /> <!--Ocean-->
          <LAYERDEF id="1" visible="true" /> <!--Countries-->
          <LAYERDEF id="2" visible="false" /> <!--States-->
          <LAYERDEF id="3" visible="false" /> <!--Provinces-->
          <LAYERDEF id="4" visible="true" /> <!--Cities-->
        </LAYERLIST>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

In the returned image, the States and Provinces layers are not included. Ocean, Countries, and Cities remain visible.



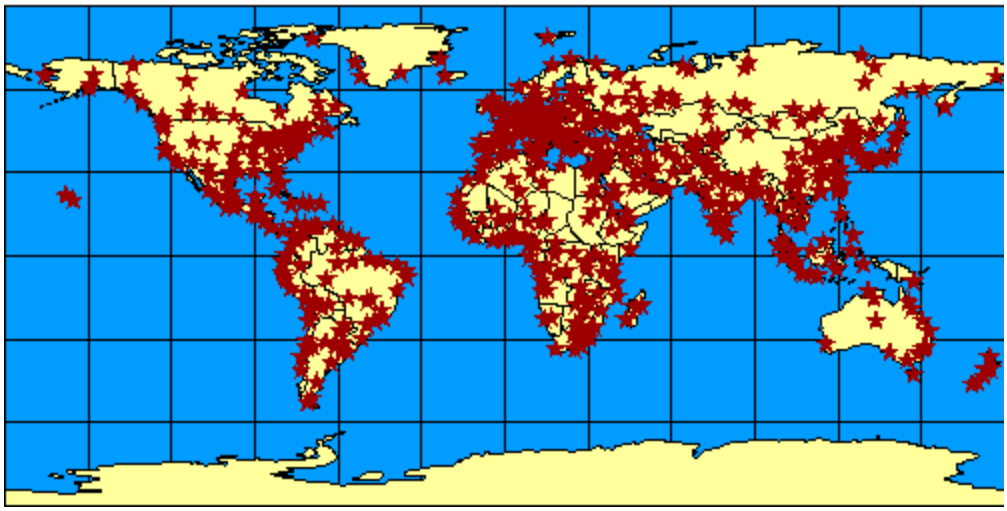
Changing layer symbology

The symbology of a layer can be changed using LAYERDEF. If no symbology information is included, then the default symbology in the Image Service is used. By adding symbology to LAYERDEF, the service information is overridden. In the next example, SIMPLERENDERER is used to change the color and symbol type for the Cities layer. The syntax for adding or modifying renderers is the same as for a map configuration file. For more information, see Using Renderers.

GET_IMAGE using LAYERDEF to change layer symbology:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <IMAGESIZE width="500" height="400" />
        <LAYERLIST>
          <LAYERDEF id="0" visible="true" />
          <LAYERDEF id="1" visible="true" />
          <LAYERDEF id="2" visible="false" />
          <LAYERDEF id="3" visible="false" />
          <LAYERDEF id="4" visible="true" >
            <SIMPLERENDERER>
              <SIMPLEMARKERSYMBOL type="star" color="155,0,0"
width="12" />
            </SIMPLERENDERER>
          </LAYERDEF>
        </LAYERLIST>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

In the returned image, the Cities layer (*id="4"*) is rendered with red stars rather than purple circles.



Querying a layer

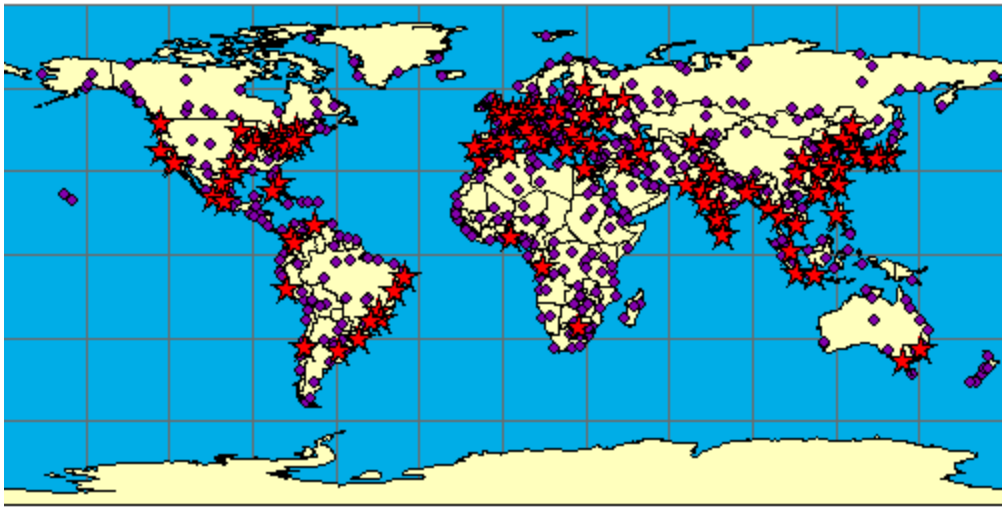
Both attribute and spatial queries can be used with LAYERDEF to set a filter on a layer. For more information on using queries and buffers, see SPATIALQUERY and BUFFER.

In the next example, the Cities layer includes an attribute query using SPATIALQUERY. The displayed cities are limited to those cities with a population greater than two million.

GET_IMAGE using an attribute query in LAYERDEF:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <IMAGESIZE width="500" height="400" />
        <LAYERLIST>
          <LAYERDEF id="0" visible="true" />
          <LAYERDEF id="1" visible="true" />
          <LAYERDEF id="2" visible="false" />
          <LAYERDEF id="3" visible="false" />
          <LAYERDEF id="4" visible="true" >
            <SPATIALQUERY where="POPULATION &gt; 2000000" />
            <SIMPLERENDERER >
              <SIMPLEMARKERSYMBOL type="star" color="155,0,0"
width="12" />
            </SIMPLERENDERER>
          </LAYERDEF>
        </LAYERLIST>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

In the returned image, only cities with a population greater than two million are rendered with red stars.

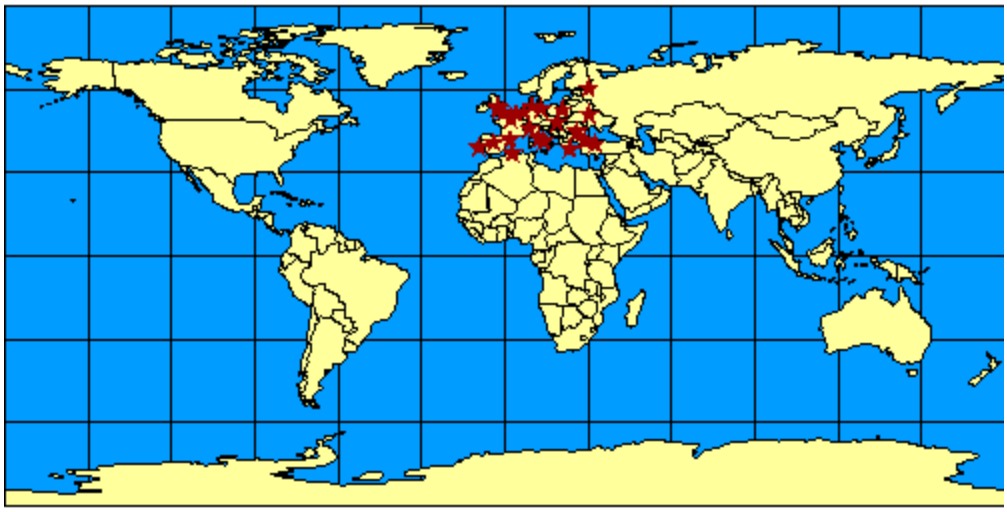


A spatial filter can also be set on a layer using SPATIALFILTER in a SPATIALQUERY. In the next example, a spatial filter is set to include only cities in Europe. In this example, ENVELOPE is used to set the filter boundary but polygons, lines, points, and buffers can also be used.

GET_IMAGE using an attribute query and spatial filter in LAYERDEF:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <IMAGESIZE width="500" height="400" />
        <LAYERLIST>
          <LAYERDEF id="0" visible="true" />
          <LAYERDEF id="1" visible="true" />
          <LAYERDEF id="2" visible="false" />
          <LAYERDEF id="3" visible="false" />
          <LAYERDEF id="4" visible="true" >
            <SPATIALQUERY where="POPULATION &gt; 2000000" >
              <SPATIALFILTER relation="area_intersection">
                <ENVELOPE minx="-14" miny="35" maxx="33" maxy="64" />
              </SPATIALFILTER>
            </SPATIALQUERY>
            <SIMPLERENDERER>
              <SIMPLEMARKERSYMBOL type="star" color="155,0,0"
width="12" />
            </SIMPLERENDERER>
          </LAYERDEF>
        </LAYERLIST>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

In the returned image, only cities with a population greater than two million within Europe are rendered with red stars.



Changing the layer order of Image Service layers

The order in which Image Service layers are drawn can be changed by using the LAYERLIST attributes *nodefault* and *order*. When the attribute *nodefault* is set to "true", only the layers listed in the LAYERLIST are displayed. Note that when *nodefault* is used, the layers are always displayed in the order in which they appear in the service, even if the order of layers is changed in the LAYERLIST.

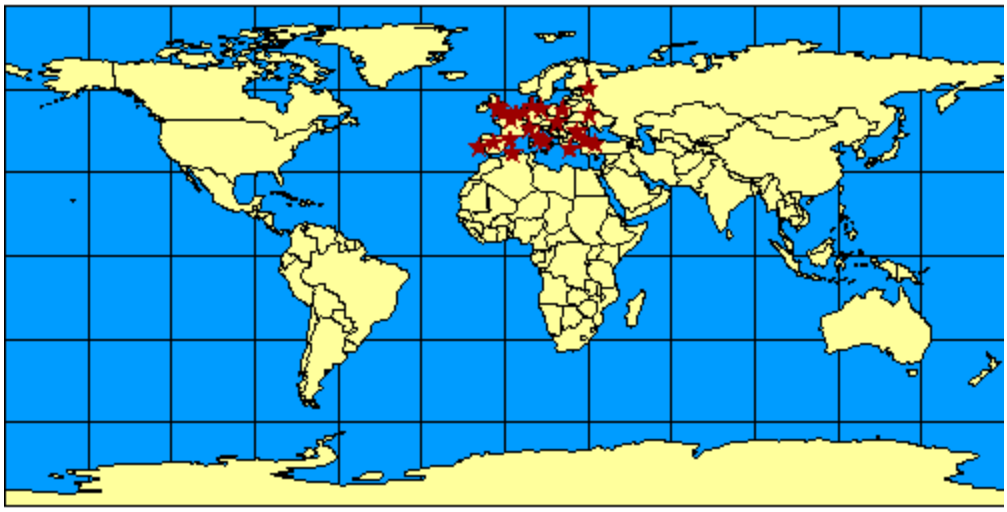
The following request has *nodefault* set to "true" in LAYERLIST. Since the LAYERDEF information for States and Provinces has been removed, they are not included in the map image. If *nodefault* were set to "false", all the layers would display even though States and Provinces are not listed.

GET_IMAGE using nodefault with LAYERLIST:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <IMAGESIZE width="500" height="400" />
        <LAYERLIST nodefault="true">
          <LAYERDEF id="0" visible="true" /> <!--Ocean-->
          <LAYERDEF id="1" visible="true" /> <!--Countries-->
          <LAYERDEF id="4" visible="true" > <!--Cities-->
            <SPATIALQUERY where="POPULATION &gt; 2000000" >
              <SPATIALFILTER relation="area_intersection">
                <ENVELOPE minx="-14" miny="35" maxx="33" maxy="64" />
              </SPATIALFILTER>
            </SPATIALQUERY>
```

```
            <SIMPLERENDERER>
              <SIMPLEMARKERSYMBOL type="star" color="155,0,0"
width="12" />
            </SIMPLERENDERER>
          </LAYERDEF>
        </LAYERLIST>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The returned image is the same as the previous example. Only cities with a population greater than two million within Europe are rendered with red stars.



The layer drawing order can be changed by using the *order* attribute in LAYERLIST. When order is set to "true", layers are drawn in the order listed in the LAYERDEF elements. Only layers in the LAYERLIST are drawn whether or not the attribute *nodefault* is used. Changing the order of the layers does not affect the order in which layers are shown in a legend.

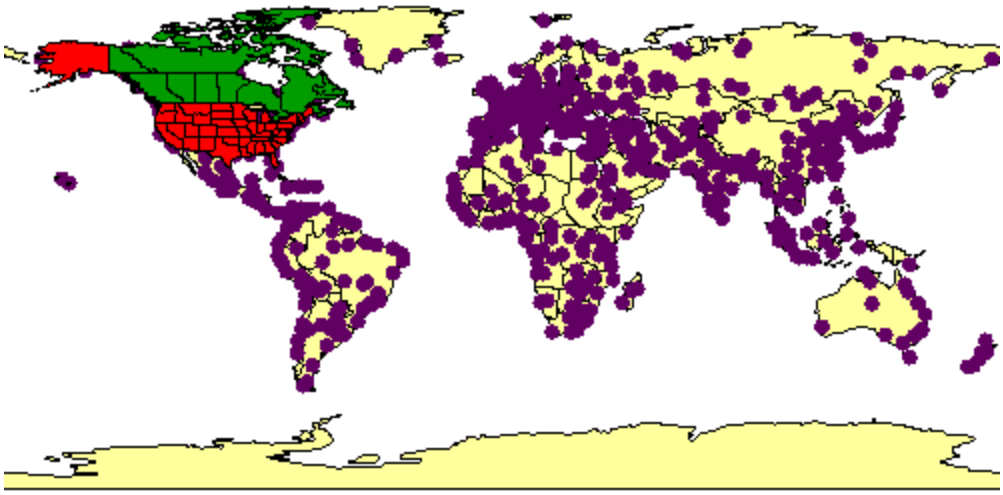
In the next example, the Cities layer (*id="4"*) is drawn before the States (*id="2"*) and Provinces (*id="3"*) layers. Also, the Ocean layer (*id="0"*) is not included in the list.

GET_IMAGE using order in LAYERLIST:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <IMAGESIZE width="500" height="400" />
        <LAYERLIST order="true">
          <LAYERDEF id="1" visible="true" /> <!--Countries-->
          <LAYERDEF id="4" visible="true" /> <!--Cities-->
          <LAYERDEF id="2" visible="true" /> <!--States-->
```

```
          <LAYERDEF id="3" visible="true" /> <!--Provinces-->
        </LAYERLIST>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

In the returned image, the Ocean layer is not included, and the States and Provinces layers are drawn on top of the Cities layer.



Adding Dynamic Layers

Dynamic layers are added to the layers already in the existing Image Service using the GET_IMAGE request. There are two general types of dynamic layers:

- For adding an existing layer in the Image Service that has been modified in some way such as showing a selected subset of data.
- For adding a new layer that is not in the service. Any LAYER type valid in a map configuration file is also valid in the GET_IMAGE request. These include point, line, polygon, image, and acetate layers.

An important consideration about dynamic layers is that they display pictures only. ***No queries or identifies can be done on dynamic layers.***

In order to add new data dynamically, the map configuration file must include *dynamic="true"* as part of the MAP element. In the map configuration file at the beginning of this document, *dynamic* is set to "true", and dynamic layers can be added in the GET_IMAGE request.

<MAP **dynamic="true"** >

This attribute does not need to be set when adding a subset of an existing layer or an

acetate layer.

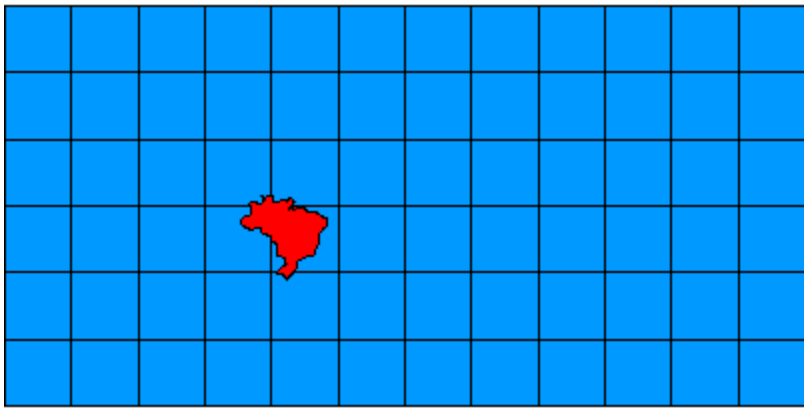
Selecting a subset of an existing layer

Subsets of existing layers in an Image Service can be included on the map as an additional layer of information. For example, assume a user selects Brazil from the Countries layer (*id="1"*) and wants to see it highlighted in red. Your first instinct may be to select Brazil by using a SPATIALQUERY in LAYERDEF such as in the following request. Note that the States, Provinces, and Cities layers have had their visibility turned off for better viewing.

Using LAYERDEF to select Brazil:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <LAYERLIST>
          <LAYERDEF id="1" type="polygon" visible="true">
            <SPATIALQUERY where="NAME=&apos;Brazil&apos;" />
            <SIMPLERENDERER>
              <SIMPLEPOLYGONSYMBOL filltype="solid"
fillcolor="255,0,0"/>
            </SIMPLERENDERER>
          </LAYERDEF>
          <LAYERDEF id="2" visible="false" />
          <LAYERDEF id="3" visible="false" />
          <LAYERDEF id="4" visible="false" />
        </LAYERLIST>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The problem with this scenario is the returned map shows only the Country of Brazil.



Although the above map shows the correct results, most people would expect to see all

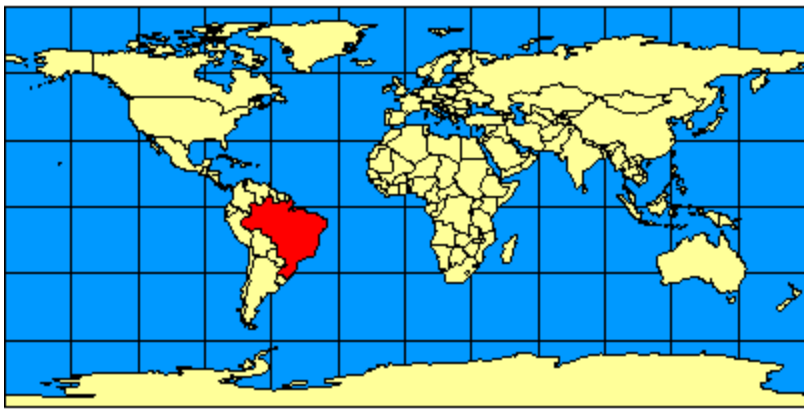
the countries with only Brazil highlighted in red. This can be done using a dynamic layer that shows Brazil as the selected country.

When setting up this type of dynamic layer, the DATASET refers back to the layer in the Image Service using its ID. In this example, DATASET *fromlayer="1"* refers to the Countries layer. The rest of the LAYER elements are used the same way as in a map configuration file. Each dynamic layer should have a unique ID that is not already used in the map configuration file or with other dynamic layers.

Using a dynamic LAYER to select Brazil:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <LAYERLIST>
          <LAYERDEF id="1" visible="true" /> <!--Countries-->
          <LAYERDEF id="2" visible="false" /> <!--States-->
          <LAYERDEF id="3" visible="false" /> <!--Provinces-->
          <LAYERDEF id="4" visible="false" /> <!--Cities-->
        </LAYERLIST>
      </PROPERTIES>
      <LAYER type="featureclass" name="Countries" visible="true"
id="10">
        <DATASET fromlayer="1" />
        <SPATIALQUERY where="NAME=&apos;Brazil&apos;" />
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0"/>
          </SIMPLERENDERER>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The resulting map shows two layers of countries. The bottom layer includes all the countries. The top layer includes only Brazil highlighted in red.



Adding new data layers dynamically

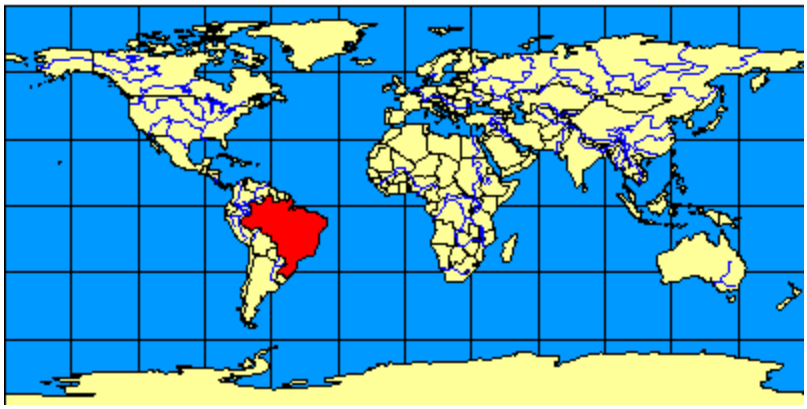
New dynamic layers can be added that include data that is not in the map configuration file. In the next example, world rivers are added as a dynamic layer. Remember that users cannot do an identify or query on this layer. The alternative is to include the layer in the Image Service and use LAYERDEF to reference it.

To add a new layer, a WORKSPACES section must be included in the map configuration file or the request. It is recommended to always include WORKSPACES in the map configuration file since references to path names on the host computer are included. By keeping all WORKSPACES in the map configuration file, the directory locations remain hidden from users. In this example, Rivers is located in the shp_ws-0 workspace, the same workspace as for Oceans, Countries, and Cities.

Using a dynamic LAYER to add a Rivers layer:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <LAYERLIST>
          <LAYERDEF id="1" visible="true" />
          <LAYERDEF id="2" visible="false" />
          <LAYERDEF id="3" visible="false" />
          <LAYERDEF id="4" visible="false" />
        </LAYERLIST>
      </PROPERTIES>
      <LAYER type="featureclass" name="Rivers" visible="true" id="20">
      <DATASET name="RIVERS" type="line" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLELINESYMBOL width="1" captype="round" color="0,0,255" />
      </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Selected Countries"
visible="true" id="10">
        <DATASET fromlayer="1" />
        <SPATIALQUERY where="NAME=&apos;Brazil&apos;" />
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0"/>
          </SIMPLERENDERER>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The resulting map shows the world rivers added as a dynamic layer.



Adding acetate layers dynamically

Acetate layers display additional features on top of a map. The acetate layers are visible only in the HTML Viewer and viewers using the ColdFusion, ActiveX, or Java Connectors, not ArcExplorer 4 or the Java Viewers.

Acetate layers are made up of one or more OBJECTs. The different objects include points, lines, polygons, text, north arrows, and scale bars. For details on the OBJECT element and children elements, refer to the Notes Section of OBJECT. Each object is placed on the acetate layer using screen coordinates or database coordinates. Screen coordinates are in pixels referenced from the bottom left-hand corner of the map frame. Database coordinates are in the coordinate system of the map.

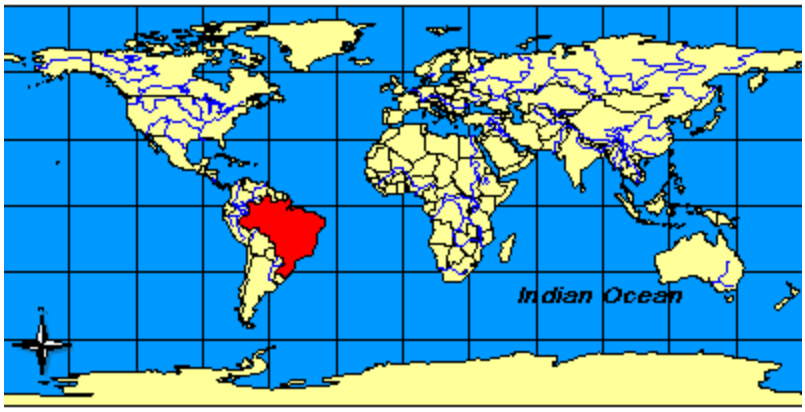
In the example below, a north arrow is added using screen coordinates. "Indian Ocean" is added as text using map coordinates.

Using a dynamic LAYER to add an acetate layer:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <LAYERLIST>
          <LAYERDEF id="0" visible="true" />
          <LAYERDEF id="1" visible="true" />
          <LAYERDEF id="2" visible="false" />
          <LAYERDEF id="3" visible="false" />
          <LAYERDEF id="4" visible="false" />
        </LAYERLIST>
      </PROPERTIES>
      <LAYER type="featureclass" name="Rivers" visible="true" id="20">
      <DATASET name="RIVERS" type="line" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLELINESYMBOL width="1" captype="round" color="0,0,255" />
      </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Selected Countries"
```

```
visible="true" id="10">
        <DATASET fromlayer="1" />
        <SPATIALQUERY where="NAME=&apos;Brazil&apos;" />
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0"/>
          </SIMPLERENDERER>
      </LAYER>
      <LAYER type="acetate" name="acetate" id="acetate">
        <OBJECT units="database">
          <TEXT coords="50 -45" label="Indian Ocean">
            <TEXTMARKERSYMBOL fontstyle="bolditalic" fontsize="12" />
          </TEXT>
        </OBJECT>
        <OBJECT units="pixel">
          <NORTHARROW type="1" size="15" coords="20 80"
shadow="32,32,32" />
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The returned map shows three dynamic layers: Brazil as a selected country, world rivers, and the north arrow and text in the acetate layer.



Layer order with dynamic layers

Dynamic layers can be drawn after, before, and within layers included in the LAYERLIST. By default, dynamic layers are drawn after the LAYERLIST layers in the same order in which they are listed in the request. In the previous example, the order is:

ID	Layer Name	Layer Type
0	Ocean	Image Service
1	Countries	Image

		Service
2	States	Image Service
3	Provinces	Image Service
4	Cities	Image Service
20	Rivers	Dynamic
10	Selected Countries	Dynamic
Acetate	Acetate	Dynamic

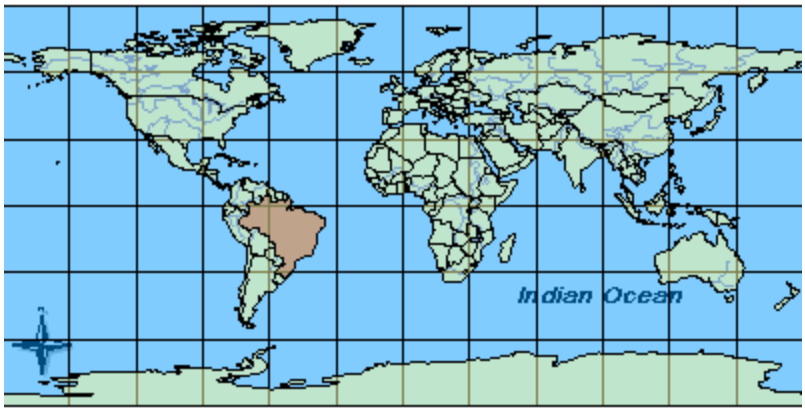
To place all dynamic layers before the LAYERLIST, the LAYERLIST attribute *dynamicfirst* can be set to true. No changes need to be made in the layer order in the request but all dynamic layers are drawn first. In the next example, *dynamicfirst* is set to "true". In order to see the dynamic layers, the transparency of the Ocean and Countries layer has been set to 50 percent (*filltransparency="0.5"*).

Drawing dynamic layers first using LAYERLIST dynamicfirst:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <LAYERLIST dynamicfirst="true" >
          <LAYERDEF id="0" type="polygon" visible="true">
            <SIMPLERENDERER>
              <SIMPLEPOLYGONSYMBOL filltype="solid"
fillcolor="0,153,255" filltransparency="0.5" />
            </SIMPLERENDERER>
          </LAYERDEF>
          <LAYERDEF id="1" type="polygon" visible="true">
            <SIMPLERENDERER>
              <SIMPLEPOLYGONSYMBOL filltype="solid"
fillcolor="255,255,153" filltransparency="0.5" />
            </SIMPLERENDERER>
          </LAYERDEF>
          <LAYERDEF id="2" visible="false" />
          <LAYERDEF id="3" visible="false" />
          <LAYERDEF id="4" visible="false" />
        </LAYERLIST>
      </PROPERTIES>
      <LAYER type="featureclass" name="Rivers" visible="true" id="20">
      <DATASET name="RIVERS" type="line" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLELINESYMBOL width="1" captype="round" color="0,0,255" />
      </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Selected Countries"
visible="true" id="10">
```

```
        <DATASET fromlayer="1" />
        <SPATIALQUERY where="NAME=&apos;Brazil&apos;" />
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0"/>
          </SIMPLERENDERER>
      </LAYER>
      <LAYER type="acetate" name="acetate" id="acetate">
        <OBJECT units="database">
          <TEXT coords="50 -45" label="Indian Ocean">
            <TEXTMARKERSYMBOL fontstyle="bolditalic" fontsize="12" />
          </TEXT>
        </OBJECT>
        <OBJECT units="pixel">
          <NORTHARROW type="1" size="15" coords="20 80"
shadow="32,32,32" />
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The returned map image shows the dynamic layers drawn first underneath the layers in LAYERLIST.



The drawing order of both Image Service and dynamic layers can be changed using the *order* attribute in LAYERLIST. If *order* is set to "true", only layers in LAYERLIST are drawn. They are drawn in the order in which they are listed in LAYERDEF, and the list can include dynamic layers. In the next example, Selected Countries and Rivers are placed before States and Provinces in the following order:

ID	Layer Name	Layer Type
0	Ocean	Image Service
1	Countries	Image Service
10	Selected	Dynamic

	Countries	
20	Rivers	Dynamic
2	States	Image Service
3	Provinces	Image Service
(4)	*(Cities)*	*(Image Service)*
Acetate	Acetate	Dynamic

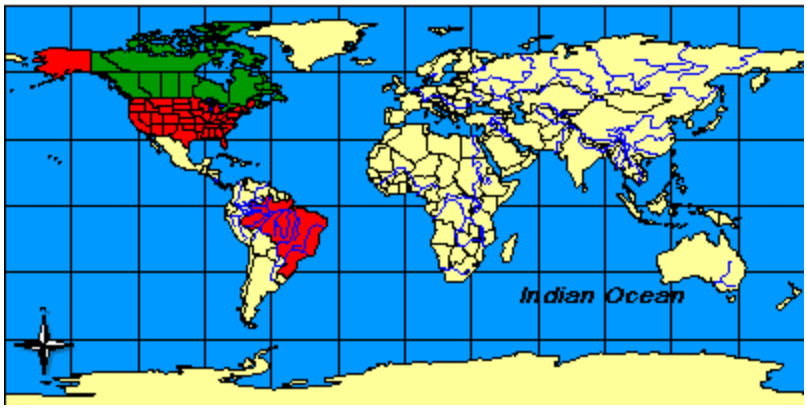
In the next example, the layers are ordered in LAYERLIST in the order in which they should appear in the image. The Cities layer is not included in the LAYERLIST and is not included in the output map image.

Using LAYERLIST order to change the order in which layers are drawn:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <LAYERLIST order="true">
          <LAYERDEF id="0" visible="true" />
          <LAYERDEF id="1" visible="true" />
          <LAYERDEF id="10" visible="true" />
          <LAYERDEF id="20" visible="true" />
          <LAYERDEF id="2" visible="true" />
          <LAYERDEF id="3" visible="true" />
          <LAYERDEF id="acetate" visible="true" />
        </LAYERLIST>
      </PROPERTIES>
      <LAYER type="featureclass" name="Rivers" visible="true" id="20">
      <DATASET name="RIVERS" type="line" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLELINESYMBOL width="1" captype="round" color="0,0,255" />
      </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Selected Countries"
visible="true" id="10">
        <DATASET fromlayer="1" />
        <SPATIALQUERY where="NAME=&amp;apos;Brazil&amp;apos;" />
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0"/>
          </SIMPLERENDERER>
      </LAYER>
      <LAYER type="acetate" name="acetate" id="acetate">
        <OBJECT units="database">
          <TEXT coords="50 -45" label="Indian Ocean">
            <TEXTMARKERSYMBOL fontstyle="bolditalic" fontsize="12" />
          </TEXT>
        </OBJECT>
        <OBJECT units="pixel">
```

```
          <NORTHARROW type="1" size="15" coords="20 80"
shadow="32,32,32" />
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The returned map image shows the dynamic layers mixed in with the Image Service layers.



Using Projections with GET_IMAGE

The projection of an image can be changed using the projection elements:

- COORDSYS
- FEATURECOORDSYS
- FILTERCOORDSYS

For a complete discussion on the different elements, refer to Using Projection Elements.

FILTERCOORDSYS is used to specify the coordinate system of the requesting client. All the examples so far have been in geographic coordinates (decimal degrees), which have an ID of "4326". All coordinates in the request, such as those for ENVELOPE, have been in geographic coordinates.

FEATURECOORDSYS is used to specify to which the coordinate system the Image Service should be transformed. In the next example, the image is requested in Robinson, which has an ID of "54030".

Using FILTERCOORDSYS and FEATURECOORDSYS:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
```

```
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <FILTERCOORDSYS id="4326" />
        <FEATURECOORDSYS id="54030" />
        <LAYERLIST>
          <LAYERDEF id="0" visible="true" />
          <LAYERDEF id="1" visible="true" />
          <LAYERDEF id="2" visible="false" />
          <LAYERDEF id="3" visible="false" />
          <LAYERDEF id="4" visible="false" />
        </LAYERLIST>
      </PROPERTIES>
      <LAYER type="featureclass" name="Rivers" visible="true" id="20">
      <DATASET name="RIVERS" type="line" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLELINESYMBOL width="1" captype="round" color="0,0,255" />
      </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Selected Countries"
visible="true" id="10">
        <DATASET fromlayer="1" />
        <SPATIALQUERY where="NAME=&apos;Brazil&apos;" />
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0"/>
          </SIMPLERENDERER>
      </LAYER>
      <LAYER type="acetate" name="acetate" id="acetate">
        <OBJECT units="database">
          <TEXT coords="50 -45" label="Indian Ocean">
            <TEXTMARKERSYMBOL fontstyle="bolditalic" fontsize="12" />
          </TEXT>
        </OBJECT>
        <OBJECT units="pixel">
          <NORTHARROW type="1" size="15" coords="20 80"
shadow="32,32,32" />
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

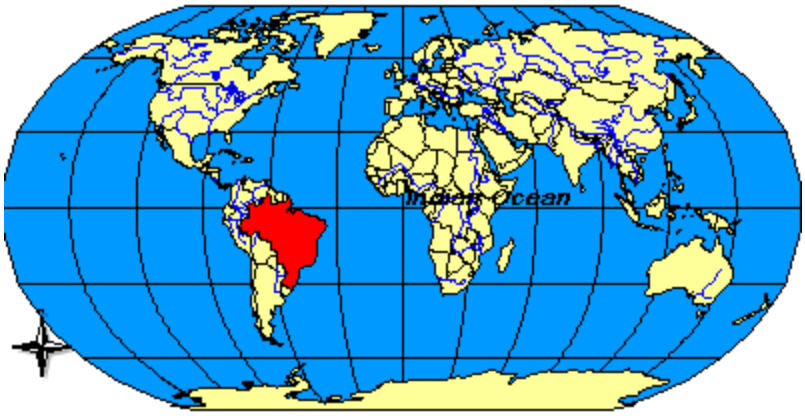
In the response, the ENVELOPE coordinates are in Robinson, which was the coordinate system used in FEATURECOORDSYS.

IMAGE response with new ENVELOPE coordinates:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-16986708.7102836" miny="-12740031.5327127"
maxx="16986708.7102836" maxy="12740031.5327127" />
      <OUTPUT file="C:\ArcIMS\output\world_MYCOMPUTER29852440.png"
url="http://mycomputer.domain.com/output/world_MYCOMPUTER29852440.png"
/>
    </IMAGE>
```

```
  </RESPONSE>
</ARCXML>
```

The returned map image shows the map in the Robinson coordinate system.



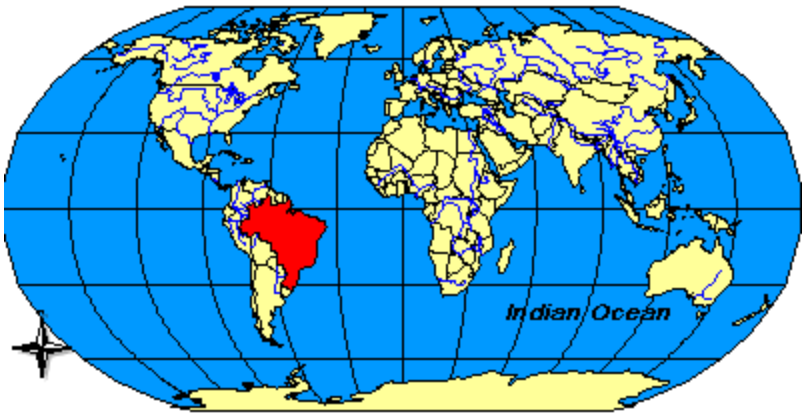
Note that the acetate OBJECT with the text "Indian Ocean" is no longer placed correctly. The problem is the OBJECT coordinate units are in geographic coordinates, while the map has been transformed to Robinson. COORDSYS can be used with an acetate OBJECT to let the ArcIMS Spatial Server know the coordinate system of the OBJECT. In the example below, COORDSYS has been added to the "Indian Ocean" object.

Including COORDSYS in the acetate layer:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <FILTERCOORDSYS id="4326" />
        <FEATURECOORDSYS id="54030" />
        <LAYERLIST>
          <LAYERDEF id="0" visible="true" />
          <LAYERDEF id="1" visible="true" />
          <LAYERDEF id="2" visible="false" />
          <LAYERDEF id="3" visible="false" />
          <LAYERDEF id="4" visible="false" />
        </LAYERLIST>
      </PROPERTIES>
      <LAYER type="featureclass" name="Rivers" visible="true" id="20">
      <DATASET name="RIVERS" type="line" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLELINESYMBOL width="1" captype="round" color="0,0,255" />
      </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Selected Countries"
visible="true" id="10">
        <DATASET fromlayer="1" />
        <SPATIALQUERY where="NAME=&apos;Brazil&apos;" />
```

```
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0"/>
          </SIMPLERENDERER>
      </LAYER>
      <LAYER type="acetate" name="acetate" id="acetate">
        <OBJECT units="database">
            <COORDSYS id="4326"/>
          <TEXT coords="50 -45" label="Indian Ocean">
            <TEXTMARKERSYMBOL fontstyle="bolditalic" fontsize="12" />
          </TEXT>
        </OBJECT>
        <OBJECT units="pixel">
          <NORTHARROW type="1" size="15" coords="20 80"
shadow="32,32,32" />
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The returned image shows the map and the "Indian Ocean" OBJECT in the Robinson coordinate system.



Using LEGEND and DRAW

LEGEND can be used in a GET_IMAGE request to include a legend for the requested map. The legend is an additional image in the same format as the map: JPG, PNG, or GIF. DRAW can be used to request only a legend rather than both a legend and a map.

The following GET_IMAGE request includes a default LEGEND since no attributes are used. DRAW has the attribute *map* set to "false" so that only a legend is returned.

GET_IMAGE request with LEGEND:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
```

```
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <LEGEND />
        <DRAW map="false" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

In the response, only the name and location of the legend are included.

IMAGE response with LEGEND:

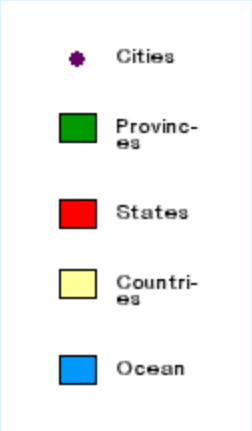
```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <LEGEND file="c:\arcims\output\world_MYCOMPUTER1248849.png"
url="http://mycomputer.domain.com/output/world_MYCOMPUTER1248849.png"
/>
    </IMAGE>
  </RESPONSE>
</ARCXML>
```



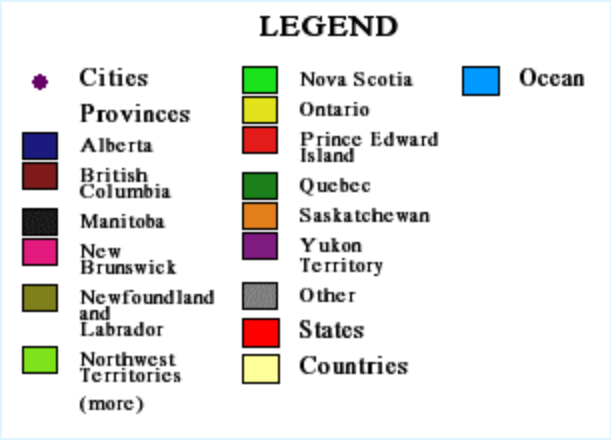
The returned legend image includes all layers that are currently visible in the map on a white background.

Several attributes can be used with LEGEND to change the look and feel of the legend image. The following table shows the results of different attribute combinations.

	When using ***reverseorder***. The layers are in the reverse order from how they are listed in the map configuration file. <LEGEND reverseorder="true" />
	When using ***backgroundcolor***. The background color can be changed using RGB values. <LEGEND backgroundcolor="255,255,0" />
	When using ***transcolor***. One transparent color can be set using RGB values. Valid only with PNG, PNG8, and GIF formats. JPEG does not support transparency. <LEGEND transcolor="255,255,255" />
	When using ***height***. A set height in pixels can be used for the legend. Note that layer information is cut off when there is too much data to fit within the height limitation. <LEGEND height="60" />
	When using ***autoextend***. *Autoextend* automatically extends the legend vertically past the specified height to accommodate all the data. If *height* is not included, the default of 300 pixels is used for the height. If data is cut off in a legend, check if autoextend is included. <LEGEND height="60" autoextend="true" />

	When using ***columns***. *Columns* defines the number of columns used in the legend. In this example, the text and swatches are close together and unreadable. This is the case since the default width of 125 pixels is used. <LEGEND height="60" autoextend="true" columns="3" />
	When using ***width***. If the legend is too narrow for the layer names, the text wraps. To avoid wrapping, increase the width of the legend. <LEGEND height="60" autoextend="true" columns="3" width="300" />
	When using ***cellspacing***. *Cellspacing* refers to the number of pixels to pad each entry in the legend. <LEGEND cellspacing="20" />
	When using ***title, font,*** and ***titlefontsize***. A title can be added to the legend. The font and font size can also be included. The *font* attribute affects all text, not just the title. <LEGEND title="LEGEND" font="Times New Roman" titlefontsize="16" />

	When using ***layerfontsize***. The layer font size can be increased or decreased. The layer font is set using the *font* attribute. The font color cannot be changed. <LEGEND title="LEGEND" font="Times New Roman" titlefontsize="16" layerfontsize="12" />
	When using ***swatchheight*** and ***swatchwidth***. The size of the swatch representing each layer can be increased or decreased using *swatchheight* and *swatchwidth*. Measurements are in pixels. <LEGEND title="LEGEND" font="Times New Roman" titlefontsize="16" swatchheight="20" swatchwidth="20" />
	When using ***cansplit*** and ***splittext***. *Cansplit* and *splittext* are valid with layers that include a VALUEMAPRENDERER. *Cansplit* allows the valuemap swatches and text to be carried over into the next column. *Splittext* is the continuation text at the bottom of the column. <LEGEND title="LEGEND" font="Times New Roman" titlefontsize="16" layerfontsize="12" columns="3" cansplit="true" splittext="(more)" width="300" autoextend="true" />



When using ***valuefontsize***. The font size of valuemap text can be controlled using *valuefontsize*. The font is set using the *font* attribute. The font color cannot be changed.

<LEGEND title="LEGEND" font="Times New Roman" titlefontsize="16" layerfontsize="12" columns="3" cansplit="true" splittext="(more)" width="330" autoextend="true" valuefontsize="10" />

Using LAYERS with LEGEND

LAYERS is used to identify which layers in a legend should NOT be included. In a GET_IMAGE request, the list can include both layers in the Image Service and dynamic layers that include DATASET. Acetate layers are never included in the legend. If LAYERS is used in a map configuration file, any LAYERS information in the request overrides the information in the map configuration file.

The following two scenarios produce the same results. In Scenario 1, a map is requested first with all the layers set to a visibility of "true". A second request is then made for a legend but no map, and the layer visibility for Ocean and Countries has been set to "false".

Scenario 1, Request 1 - GET_IMAGE using LAYERDEF to set layer visibility to "true" for map:

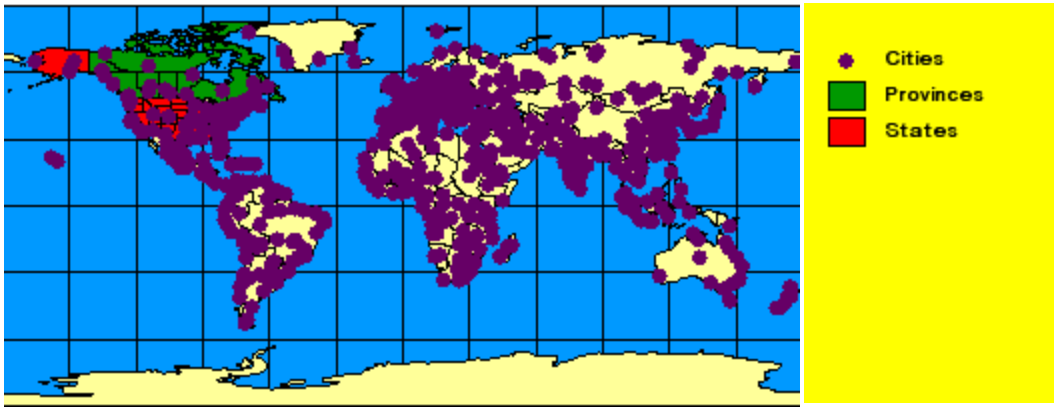
```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <IMAGESIZE width="500" height="400" />
        <LAYERLIST>
          <LAYERDEF id="0" visible="true" />
          <LAYERDEF id="1" visible="true" />
          <LAYERDEF id="2" visible="true" />
          <LAYERDEF id="3" visible="true" />
          <LAYERDEF id="4" visible="true" />
        </LAYERLIST>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
```

```
</ARCXML>
```

Scenario 1, Request 2 - GET_IMAGE using LAYERDEF to set the visibility of some layers to "false" for legend:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <LAYERLIST>
          <LAYERDEF id="0" visible="false" />
          <LAYERDEF id="1" visible="false" />
          <LAYERDEF id="2" visible="true" />
          <LAYERDEF id="3" visible="true" />
          <LAYERDEF id="4" visible="true" />
        </LAYERLIST>
        <LEGEND title="LEGEND" backgroundcolor="255,255,0" />
        <DRAW map="false" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Two images are returned in two separate IMAGE responses. In the first response, the map name and location is sent. In the second response, the legend name and location are sent. In the images, the map contains all layers, while the legend does not include Ocean or Countries.



In Scenario 2, LAYERS is used to list which layers should not be included in the legend. The advantage of using LAYERS is that both a map and a legend can be requested at the same time. The following request produces the same results as the two previous requests in Scenario 1.

Using LAYERS in a GET_IMAGE request:

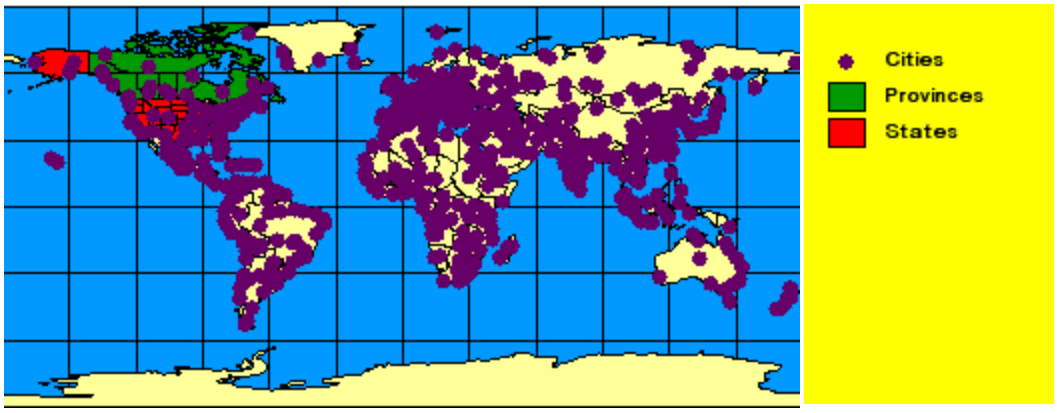
```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
```

```
    <GET_IMAGE>
      <PROPERTIES>
        <LEGEND backgroundcolor="255,255,0" >
          <LAYERS>
            <LAYER id="0"/>
            <LAYER id="1"/>
          </LAYERS>
        </LEGEND>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Both the map with all layers and legend with selected layers are returned in one response.

IMAGE response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-180" miny="-144" maxx="180" maxy="144" />
      <OUTPUT file="c:\arcims\output\world_MYCOMPUTER2102209.png"
url="http://mycomputer.domain.com/output/world_MYCOMPUTER2102209.png"
/>
      <LEGEND file="c:\arcims\output\world_MYCOMPUTER1248849.png"
url="http://mycomputer.domain.com/output/world_MYCOMPUTER1248849.png"
/>
    </IMAGE>
  </RESPONSE>
</ARCXML>
```



Dynamic layers in a legend

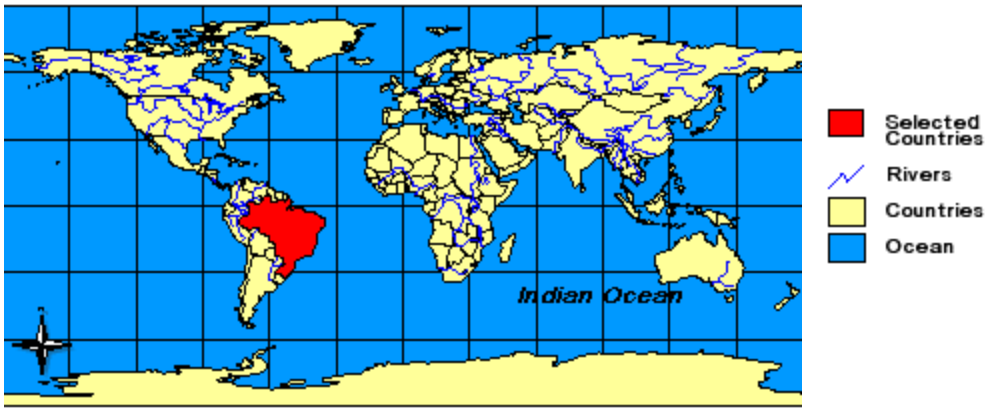
Dynamic layers that contain DATASET are included in the legend. Acetate layers are never included in the legend. In the following example, a simple LEGEND is requested.

Including LEGEND with dynamic layers:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <LAYERLIST>
          <LAYERDEF id="1" visible="true" />
          <LAYERDEF id="2" visible="false" />
          <LAYERDEF id="3" visible="false" />
          <LAYERDEF id="4" visible="false" />
        </LAYERLIST>
        <LEGEND />
      </PROPERTIES>
      <LAYER type="featureclass" name="Rivers" visible="true" id="20">
      <DATASET name="RIVERS" type="line" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLELINESYMBOL width="1" captype="round" color="0,0,255" />
      </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Selected Countries"
visible="true" id="10">
        <DATASET fromlayer="1" />
        <SPATIALQUERY where="NAME=&apos;Brazil&apos;" />
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0"/>
          </SIMPLERENDERER>
      </LAYER>
      <LAYER type="acetate" name="acetate" id="acetate">
        <OBJECT units="database">
          <TEXT coords="50 -45" label="Indian Ocean">
            <TEXTMARKERSYMBOL fontstyle="bolditalic" fontsize="12" />
          </TEXT>
        </OBJECT>
        <OBJECT units="pixel">
          <NORTHARROW type="1" size="15" coords="20 80"
shadow="32,32,32" />
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The returned map shows Ocean, Countries, Selected Countries, Rivers, and the acetate layer. All but the acetate layer are included in the legend.



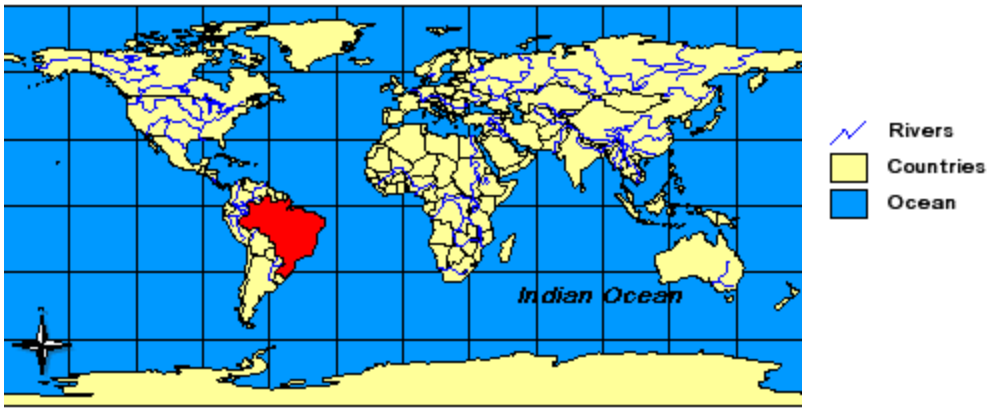
Dynamic layers can also be suppressed in the legend by using LAYERS. In the next example, the "Selected Countries" layer (*id="10"*) is not included.

Using LAYERS with dynamic layers:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <LAYERLIST>
          <LAYERDEF id="0" visible="true" />
          <LAYERDEF id="1" visible="true" />
          <LAYERDEF id="2" visible="false" />
          <LAYERDEF id="3" visible="false" />
          <LAYERDEF id="4" visible="false" />
        </LAYERLIST>
        <LEGEND>
          <LAYERS>
            <LAYER id="10"/>
          </LAYERS>
        </LEGEND>
      </PROPERTIES>
      <LAYER type="featureclass" name="Rivers" visible="true" id="20">
      <DATASET name="RIVERS" type="line" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLELINESYMBOL width="1" captype="round" color="0,0,255" />
      </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Selected Countries"
visible="true" id="10">
        <DATASET fromlayer="1" />
        <SPATIALQUERY where="NAME=&apos;Brazil&apos;" />
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0"/>
          </SIMPLERENDERER>
      </LAYER>
      <LAYER type="acetate" name="acetate" id="acetate">
        <OBJECT units="database">
          <TEXT coords="50 -45" label="Indian Ocean">
```

```
            <TEXTMARKERSYMBOL fontstyle="bolditalic" fontsize="12" />
          </TEXT>
        </OBJECT>
        <OBJECT units="pixel">
          <NORTHARROW type="1" size="15" coords="20 80"
shadow="32,32,32" />
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The returned map is the same as the previous example. However, the legend does not include the "Selected Countries" layer.



Using OUTPUT to Control Image Names and Locations

OUTPUT, when used in a GET_IMAGE request, defines the location and filename of the output map and legend image files. By default, the image output directory and URL location are determined at the time the Image Service is started. When a request is made, the ArcIMS Spatial Server assigns a filename. The name includes the Image Service name, the computer name the image was generated on, and a randomly generated number. If the service is named "world" and the computer is "MYCOMPUTER", then an example filename is "world_MYCOMPUTER1248849.png". The ArcIMS Tasker Windows service or UNIX daemon automatically deletes these generated images on a user-specified interval.

OUTPUT can be used in both a map configuration file and in a request. When used in a map configuration file, the OUTPUT information overrides the output information stored when the service is started. When used in a request, OUTPUT overrides information in both the map configuration file and when the service is started. When OUTPUT is used, the output files are not automatically deleted by ArcIMS Tasker. In order for the files to be deleted, the *taskfile* property must be set in tasker.properties. For information on setting this property, see the topic "Setting up ArcIMS Monitor and ArcIMS Tasker Property Files" in the System Administration: Administering Property Files chapter of the ArcIMS Help Contents.

OUTPUT works with paired attributes. If one of the attributes is used, its pair is required. There are attribute pairs for both the map and legend images. Determining which pair to use depends on whether the ArcIMS Spatial Server defines the output filename or the user does. In all cases, the output directory and URL are user-specified.

Attribute	Paired Attribute	Filename Assignment	Example: http://mycomputer/arcims/...
path	baseurl	ArcIMS assigns random filename.	world_MYCOMPUTER1248849.png
name	url	User assigns a filename.	myfilename.png
legendpath	legendbaseurl	ArcIMS assigns random filename.	world_MYCOMPUTER1248851.png
legendname	legendurl	User assigns a filename.	myfilename.png

When starting an Image Service or using OUTPUT in a request, UNC pathnames are valid. For example, instead of using "c:\arcims\output", "\\myComputer\arcims\output" can be used.

ArcIMS spatial server assigns filename

In the following example, the ArcIMS Spatial Server assigns a filename but the user assigns the directory and URL for both the map and the legend. The map and the legend images do not need to be written to the same directory. The attribute pairs in this scenario are *path-baseurl* and *legendpath-legendbaseurl*.

Note that even though *legendpath* and *legendbaseurl* are included in OUTPUT, the LEGEND element *must* be present to generate a legend image. If LEGEND is not included, no legend is generated. LEGEND can be included in either the map configuration file or the request.

OUTPUT when ArcIMS Spatial Server defines filename:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <OUTPUT path="c:\arcims"
baseurl="http://mycomputer.domain.com/arcims"
legendpath="c:\arcims\legend"
legendbaseurl="http://mycomputer.domain.com/arcims/legend" />
```

```
        <LEGEND />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

IMAGE response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-180" miny="-144" maxx="180" maxy="144" />
      <OUTPUT file="c:\arcims\world_MYCOMPUTER2983738.png"
url="http://mycomputer.domain.com/world_MYCOMPUTER2983738.png" />
      <LEGEND file="c:\arcims\legend\world_MYCOMPUTER2983739.png"
url="http://mycomputer.domain.com/output/world_MYCOMPUTER2983739.png"
/>
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

User assigns filename

In the next example, the user assigns a filename, directory, and URL for both the map and the legend. The attribute pairs in this scenario are *name-url* and *legendname-legendurl*. Once again, to generate a legend, LEGEND must be included in either the map configuration file or the request.

OUTPUT when user defines filename:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <OUTPUT name="c:\arcims\mapimage.png"
url="http://mycomputer.domain.com/arcims/mapimage.png"
legendname="c:\arcims\legend\legendimage.png"
legendurl="http://mycomputer.domain.com/arcims/legend/legendimage.png"
/>
        <LEGEND />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

IMAGE response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
```

```
      <ENVELOPE minx="-180" miny="-144" maxx="180" maxy="144" />
      <OUTPUT file="c:\arcims\mapimage.png"
url="http://mycomputer.domain.com/mapimage.png" />
      <LEGEND file="c:\arcims\legend\legendimage.png"
url="http://mycomputer.domain.com/legend/legendimage.png" />
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

Restricting OUTPUT

When using an ArcIMS HTML Viewer, ArcIMS Java Viewer, ArcExplorer 4, or any other browser using the ArcIMS Servlet Connector, the OUTPUT element is restricted by default.

- In a GET_IMAGE request, OUTPUT is ignored.

GET_IMAGE request when OUTPUT is restricted:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <OUTPUT name="c:\arcims\mapimage.png"
url="http://mycomputer.domain.com/arcims/mapimage.png" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

- In an IMAGE response, the attribute *file* is not returned. The OUTPUT location is the default location specified when the Image Service was started. The ArcIMS Spatial Server assigns the filename. OUTPUT includes only *url*.

IMAGE response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-180" miny="-144" maxx="180" maxy="144" />
      <OUTPUT
url="http://mycomputer.domain.com/output/world_MYCOMPUTER2102209.png"
/>
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

These restrictions can be lifted by setting the properties *spatialServer.AllowRequestOutput* and *spatialServer.AllowResponsePath* to "true" in

esrimap_prop. This property file is found in the servlet engine directory. For more information on the location of esrimap_prop and its properties, see the topic "Administering the ArcIMS Servlet Connector" in the System Administration chapter of the Using ArcIMS Help Contents.

These restrictions apply only when the ArcIMS Servlet Connector is used. They do not apply to the ActiveX, ColdFusion, or Java Connectors. The OUTPUT filename extensions are restricted to *.jpg, *.png, or *.gif, regardless of whether OUTPUT is restricted or not.

Using GET_IMAGE and IMAGE with ArcMap Image Services

Introduction

This document covers GET_IMAGE and IMAGE when using ArcMap Image Services. When using Image Services, see Using GET_IMAGE and IMAGE with Image Services.

The purpose of GET_IMAGE and IMAGE is to render a map image on the ArcIMS Spatial Server and provide the location and filename of that image. GET_IMAGE is for generating a map only. To retrieve attribute data associated with the map, a separate GET_FEATURES request must be made.

ArcMap documents contain one or more data frames. The data frames can be viewed as a map or in a layout. For infomation on using layouts, see Using GET_LAYOUT and LAYOUT with ArcMap Image Services. Most of the sample requests that follow are based on an ArcMap document (*.mxd) consisting of two data frames called Layers and States. The Layers data frame has five layers from the ESRIDATA dataset. The following table summarizes the layer names, shapefile name, and layer ID number.

Layers data frame Layer Name	Shapefile Name	Layer ID
Cities	CITIES	0
Provinces	PROVINCE	1
States	STATES	2
Countries	CNTRY94	3
Ocean	WORLD30	4

The second data frame is used in some examples. This data frame is named States and contains the following layers:

States data frame Layer Name	Shapefile Name	Layer ID
Cities	CITIES	0
Roads	ROADS	1
States	STATES	2

GET_IMAGE Request and IMAGE Response

The simplest GET_IMAGE request includes PROPERTIES with no child elements inside. With this request, a map image is generated using information established in the ArcMap Image Service.

Simple GET_IMAGE request:

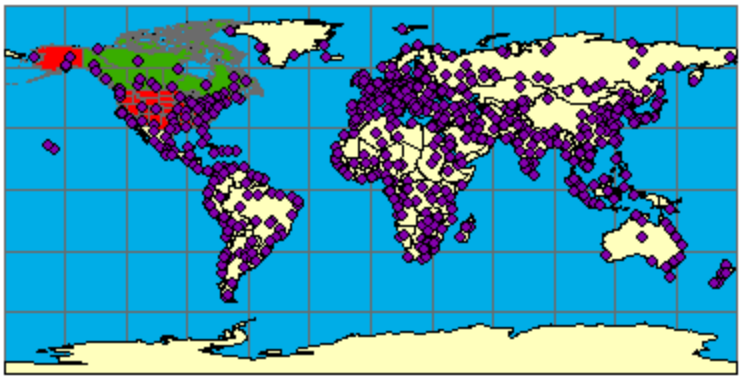
```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The IMAGE response includes a default envelope and the name and location of the generated map image.

IMAGE response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-180" miny="-144" maxx="180" maxy="144" />
      <OUTPUT
url="http://mycomputer.domain.com/output/world_MYCOMPUTER2102209.png"
file="c:\arcims\output\world_MYCOMPUTER2102209.png" />
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

The returned map image is generated using defaults from the service. The visible layers shown are Ocean, Countries, States, Provinces, and Cities. Also, the map is in geographic coordinates, and the default envelope includes the entire world.



Using GET_IMAGE dataframe

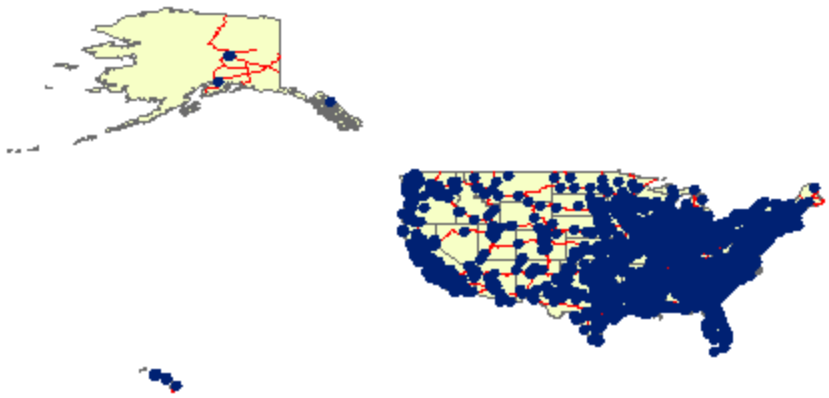
ArcMap documents contain one or more data frames. By default, a GET_IMAGE request accesses the active data frame. To gain access to a different data frame, the attribute

dataframe is used. In the next example, the attribute *dataframe="States"* is included with GET_IMAGE.

GET_IMAGE request with dataframe="States":

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE dataframe="States">
      <PROPERTIES>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The returned map image includes the States data frame. The layers shown are States, Roads, and Cities. The map is in geographic coordinates, and the default envelope includes the entire United States.



Using GET_IMAGE autoresize

During a GET_IMAGE request, the maximum size of an image can be no greater than the image memory limit set when an ArcMap Image Service is started. For example, an image memory limit of 1 MB allows a map no larger than 262,144 pixels (512 x 512) to be generated. By default, an image is limited in size to 4 MB, which corresponds to 1,048,576 pixels. Changes can be made to the image size and pixel count using the ArcIMS Administrator. For more information, see *Using ArcIMS*.

If *autoresize* is set to "true" in GET_IMAGE, a requested map greater than the maximum pixel count is reduced in size to within the maximum pixel count. In the next example, the service image limit is 4 MB. IMAGESIZE requests an image greater than 4 MB. (IMAGESIZE is discussed in greater detail in the IMAGESIZE section).

GET_IMAGE request with autoresize="true":

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE autoresize="true">
      <PROPERTIES>
        <IMAGESIZE width="2000" height="1600" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The IMAGE response includes a resized image, because the requested image size was greater than allowed. In OUTPUT, the attributes *height* and *width* show the new image size. (OUTPUT is discussed in greater detail in the OUTPUT section).

IMAGE response with resized image information:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-220.916666666667" miny="-176.706042824789"
maxx="220.916666666667" maxy="176.706042824789" />
      <OUTPUT
url="http://mycomputer.domain.com/output/world_MYCOMPUTER3633699.jpg
width="1619" height="1295"
file="c:\arcims\output\world_MYCOMPUTER3633699.jpg" />
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

If *autoresize* is "false" or not included in the GET_IMAGE request, and the requested image is too big, an error message is returned.

IMAGE response when requested map is too large:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <ERROR machine="MYMACHINE" processid="1324"
threadid="1904">Requested image is too big and can not be
created</ERROR>
  </RESPONSE>
</ARCXML>
```

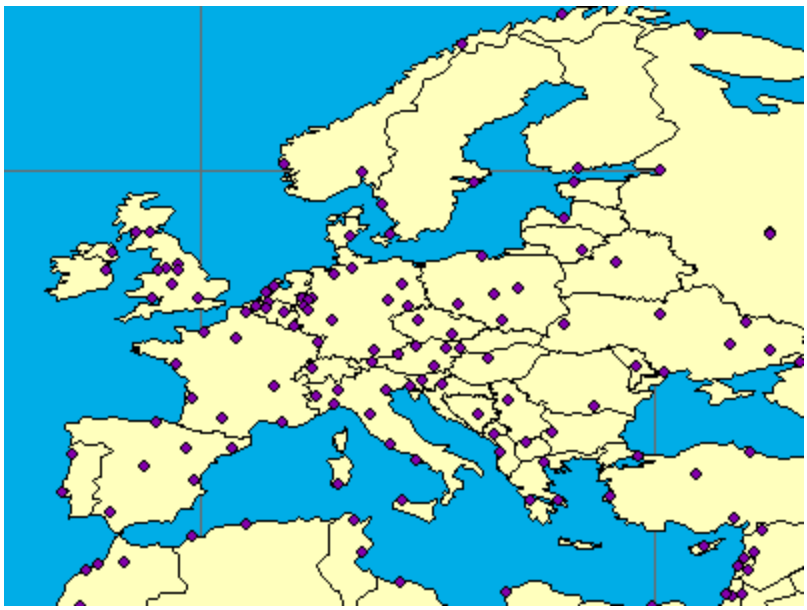
Zoom and Pan with ENVELOPE

When a user pans or zooms on a map, what really happens is the map extent changes. In a GET_IMAGE request, the extent is changed by sending new x and y minimum and maximum coordinates in an ENVELOPE. This new envelope overrides the envelope set in the service. In the following example, the envelope zooms to the region around Europe.

GET_IMAGE request with a change in ENVELOPE:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-13" miny="37" maxx="40" maxy="65" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The returned map image includes only Europe.



Changing the Output IMAGESIZE

IMAGESIZE sets the size of the output map image in pixels. If IMAGESIZE is not used in a request, the default image size is 400 x 300 pixels. As noted in the Using Autoresize section, the maximum size of an image can be no greater than the image memory limit set when a service is started.

The output image sizes can be controlled in two ways using the following attribute groups:

- *width* and *height*
- *width*, *height*, and *dpi*

The following example, using a New York City street service, shows a GET_IMAGE request with IMAGESIZE. The *width* and *height* are set to 250 and 175 pixels, respectively.

GET_IMAGE request using IMAGESIZE:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-73.985" miny="40.756" maxx="-73.972"
maxy="40.765" />
        <IMAGESIZE width="250" height="175"/>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

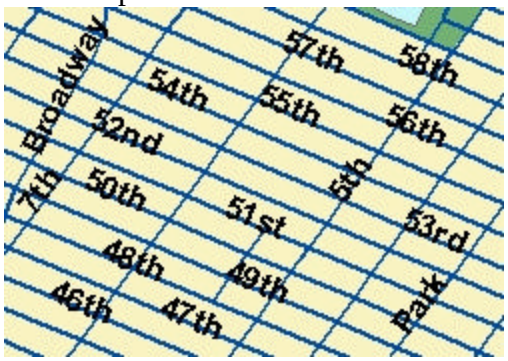
Depending on the combination of IMAGESIZE attributes and ENVELOPE size used in the request, the results of the returned map image may differ.

IMAGESIZE width and height

Width and *height* are required attributes. They are used to draw a map at the specified width and height based on the envelope. If *width* and *height* are changed but the envelope remains the same, the scale of the map changes. In other words, if *width* and *height* are increased, the map is larger and, in effect, "zoomed in". If a scale threshold is met, a layer might be added or removed.

In the two images below, a set of streets in New York City is shown. The first map is 250 x 175 pixels in size. The second map has the same envelope but is 350 x 250 pixels. Note that symbology has changed for the streets. By making the map larger and zooming in, a scale threshold was met instructing the ArcIMS Spatial Server to change to a layer with different symbology.

250x175 pixels:




```
<PROPERTIES>
  <ENVELOPE minx="-73.985"
miny="40.756" maxx="-73.972"
maxy="40.765" />
  <IMAGESIZE width="250" height="175"
/>
</PROPERTIES>
```

350x250 pixels:




```
<PROPERTIES>
  <ENVELOPE minx="-73.985"
miny="40.756" maxx="-
73.972" maxy="40.765" />
  <IMAGESIZE width="350"
height="250" />
</PROPERTIES>
```

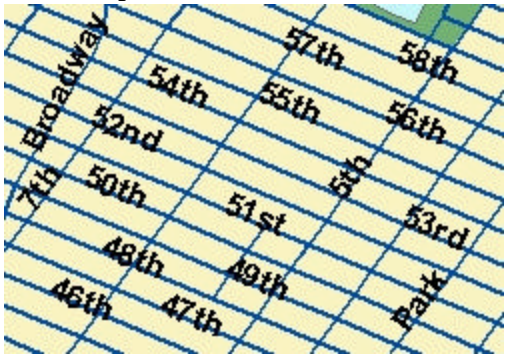
IMAGESIZE width, height, and dpi

To avoid having layers added or removed when the map size is increased or decreased, the attribute *dpi* can be used along with *width* and *height*.

The next examples use *width*, *height*, and *dpi*. The first image is for reference and includes only *height* and *width*. In the second image, the *width* and *height* values are changed to the desired size of the output image, in this case 350 and 245 pixels, respectively. In order to keep the scales proportionate, a new dpi must be calculated. The formula is (new *width*) / (original *width*) * (*dpi* of the service). If *dpi* is not in the service, a default value of 96 is used. In this example, assuming a service dpi of 96, the new *dpi* is (350 / 250 * 96) or 134. The *dpi* is then set to 134.

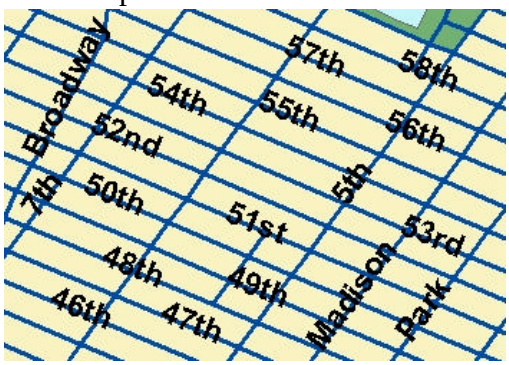
Note that as the map size increases or decreases, the symbols scale proportionally. Whether or not symbols scale is determined at the time the ArcMap document is created.

250x175 pixels:




```
<PROPERTIES>
  <ENVELOPE minx="-73.985"
miny="40.756" maxx="-73.972"
maxy="40.765" />
  <IMAGESIZE width="250" height="175"
/>
</PROPERTIES>
```

350x245 pixels:




```
<PROPERTIES>
  <ENVELOPE minx="-73.985"
miny="40.756" maxx="-73.972"
maxy="40.765" />
  <IMAGESIZE width="350" height="245"
dpi="134" />
</PROPERTIES>
```

Using dpi to assure correct scale thresholds

The attribute *dpi* can be used in IMAGESIZE, not just to accommodate new image sizes, but also to assure that scale thresholds are calculated correctly for images that do not change size. An ArcMap document can be generated using many different screen sizes. Based on the screen size, the dpi is different, and scale thresholds will differ from machine to machine. A dpi of 96 is common for screens that are 1024 x 768 pixels.

The clients can also have a variety of dpi values. If a client dpi is different from the service dpi, a map will have different scale thresholds on the client. The differences are often small but sometimes noticeable.

The ArcIMS Java Viewers, ArcExplorer 4, and ArcMap can determine the dpi of a client by making system calls. These clients can include *dpi* in IMAGESIZE. All layer scale thresholds are recalculated each time a map is generated. The HTML Viewers, on the other hand, cannot make system calls, and therefore cannot calculate the client dpi. With the ArcIMS HTML Viewer, a dpi of 96 is assumed. If the HTML Viewer is displayed on a screen other than 96 dpi, the scale dependencies will behave differently.

Using BACKGROUND

BACKGROUND is used to define a color for the image background. It can also be used to make one color in the image transparent. Depending on the browser, the image formats that support transparent colors vary. JPG images do not support transparent colors. The table below lists which image formats support transparent colors for different browsers. Note that the ArcMap Server does not support the GIF format for image output.

Browser	Supported Transparent Image Formats

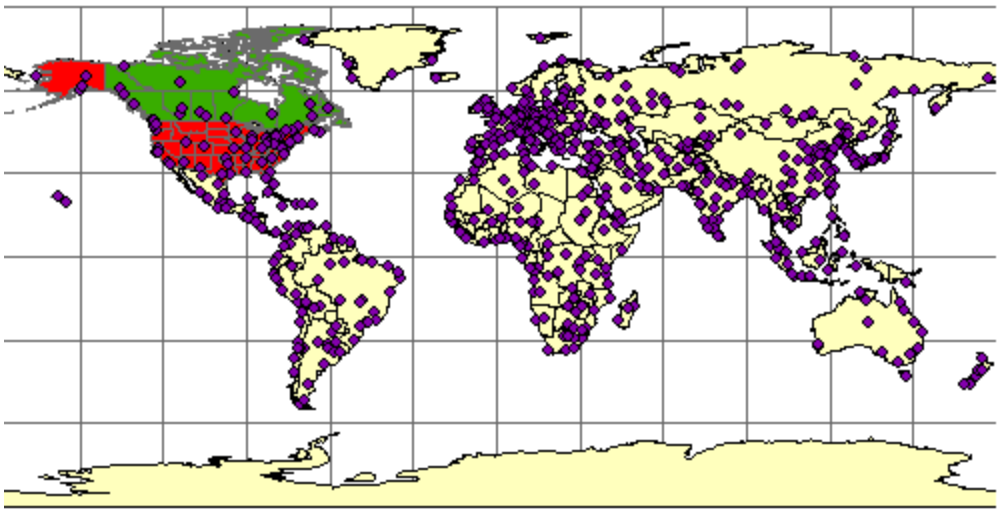
ArcIMS HTML Viewer in Internet Explorer	PNG8
ArcIMS HTML Viewer in Netscape 4.x	None
ArcIMS HTML Viewer in Netscape 6.x	PNG8
ArcExplorer 4	PNG24, PNG8
ArcIMS Java Viewers in Internet Explorer and Netscape	PNG24, PNG8

To make a color transparent, both the *color* and *transcolor* attributes of BACKGROUND must be set to the same color. In the following request, the transparent color is the blue color in the Ocean layer.

GET_IMAGE request using BACKGROUND:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <IMAGESIZE width="500" height="400" />
        <BACKGROUND color="0,153,255" transcolor="0,153,255" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

In the returned image, the ocean is now transparent.



Using LAYERLIST and LAYERDEF

LAYERLIST and LAYERDEF are used together in a GET_IMAGE request to toggle layer visibility on and off.

Using LAYERDEF visible to set layer visibility

Layers in an Image Service can be switched on and off using the attribute *visible* in LAYERDEF. If LAYERDEF is not included, the layer visibility is set to the visibility of the layers in the Image Service. Within the LAYERDEF element, layers that are set to *visible="false"* are not included in the image.

The layers in LAYERDEF are identified by their ID. The following table lists which ID corresponds to which layer.

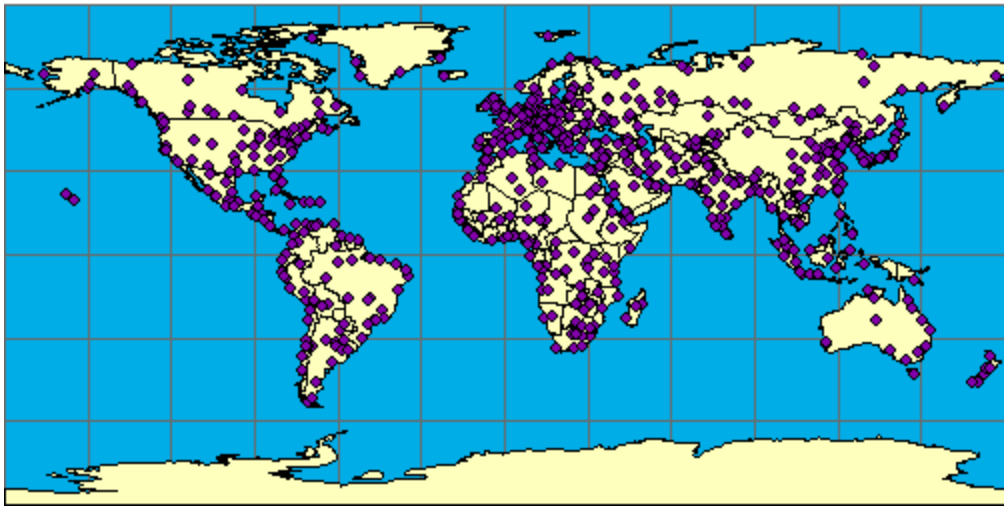
ID	Layer Name
0	Cities
1	Provinces
2	States
3	Countries
4	Ocean

In the following example, the States and Provinces layers have their visibility set to "false".

GET_IMAGE using LAYERDEF to set layer visibility:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <IMAGESIZE width="500" height="400" />
        <LAYERLIST>
          <LAYERDEF id="4" visible="true" /> <!--Ocean-->
          <LAYERDEF id="3" visible="true" /> <!--Countries-->
          <LAYERDEF id="2" visible="false" /> <!--States-->
          <LAYERDEF id="1" visible="false" /> <!--Provinces-->
          <LAYERDEF id="0" visible="true" /> <!--Cities-->
        </LAYERLIST>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

In the returned image, the States and Provinces layers are not included. Ocean, Countries, and Cities remain visible.



Using LAYERLIST order to set layer visibility

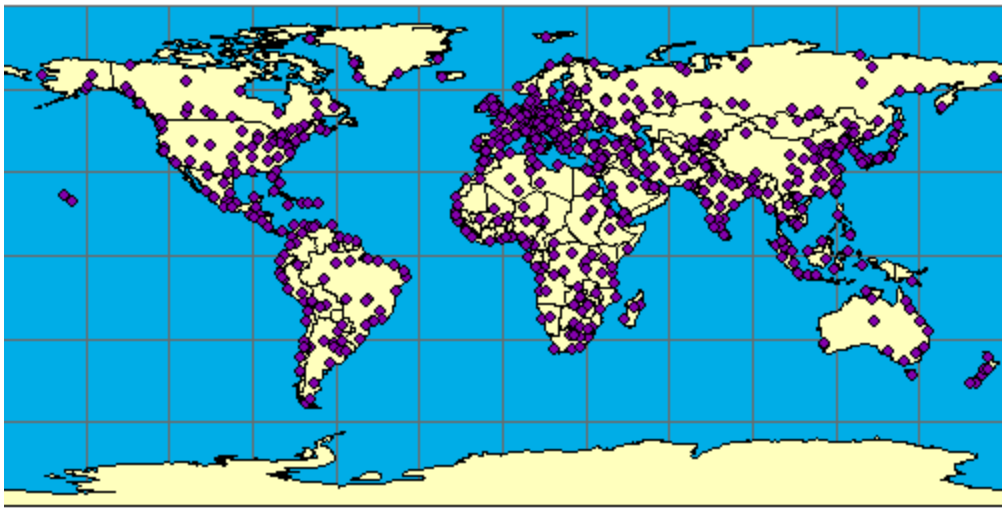
A second way to set the visibility of a layer is to use the attribute *order* in LAYERLIST. When *order* is used, only the layers listed in the LAYERLIST are included in the output. Note that with ArcMap Image Services, the layers are always displayed in the order in which they appear in the service, even if the order of layers is changed in LAYERLIST.

The following request has *order* set to "true" in LAYERLIST. Since the LAYERDEF information for States and Provinces has been removed, they are not included in the map image.

GET_IMAGE using order with LAYERLIST:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <IMAGESIZE width="500" height="400" />
        <LAYERLIST order="true">
          <LAYERDEF id="4" visible="true" /> <!--Ocean-->
          <LAYERDEF id="3" visible="true" /> <!--Countries-->
          <LAYERDEF id="0" visible="true" /> <!--Cities-->
        </LAYERLIST>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The returned image does not include the States or Provinces layers but does include Ocean, Countries, and Cities.



Adding Information in a Request

An ArcMap Image Service has a defined number of layers in the service. Unlike Image Services, new data cannot be added as a dynamic layer. However, selected data sets and acetate layers can both be added in a GET_IMAGE request.

Selecting a subset of an existing layer

Subsets of existing layers in a service can be included on a map. For example, assume a user selects Brazil from the Countries layer (id="3"). A request can be written two ways to yield the same results.

Method 1: Including a query in LAYER. When setting up this type of query, a LAYER is added to the GET_IMAGE request. DATASET must be included, and the *fromlayer* value refers back to the LAYER *id* in the service.

Both attribute and spatial queries can be used with LAYER to set a filter on a layer. For more information on using queries and buffers, see SPATIALQUERY, SPATIALQUERY, and BUFFER.

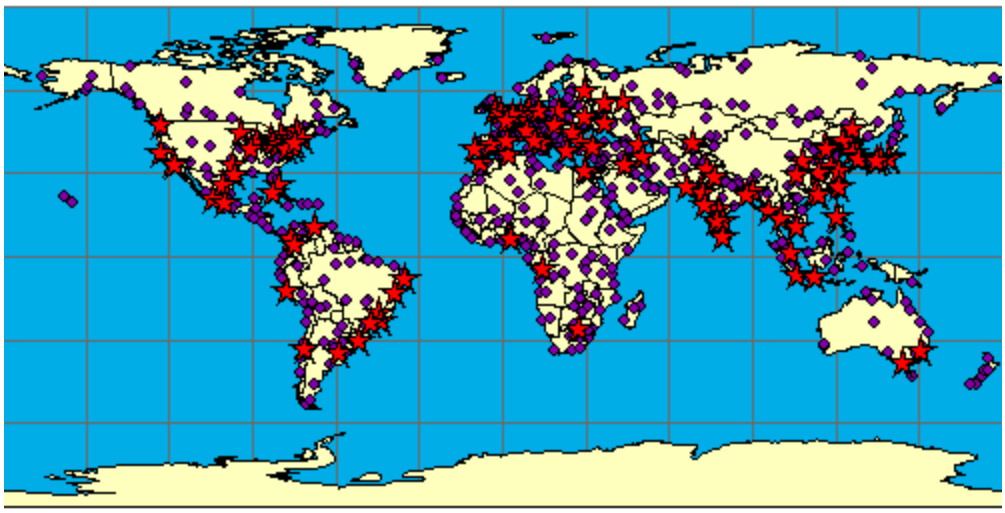
In the next example, the Cities layer includes an attribute query using SPATIALQUERY. The displayed cities are limited to those cities with a population greater than two million.

GET_IMAGE using an attribute query in LAYER:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <IMAGESIZE width="500" height="400" />
        <LAYERLIST>
```

```
          <LAYERDEF id="4" visible="true" />
          <LAYERDEF id="3" visible="true" />
          <LAYERDEF id="2" visible="false" />
          <LAYERDEF id="1" visible="false" />
          <LAYERDEF id="0" visible="true" />
        </LAYERLIST>
      </PROPERTIES>
      <LAYER type="featureclass" name="Cities" visible="true" id="10">
        <DATASET fromlayer="0" />
        <SPATIALQUERY where="POPULATION &gt; 2000000" />
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

In the returned image, cities with a population greater than two million are rendered with the selection symbol defined in the ArcMap document. Unlike Image Services, selection symbols cannot be changed using renderers in the request.



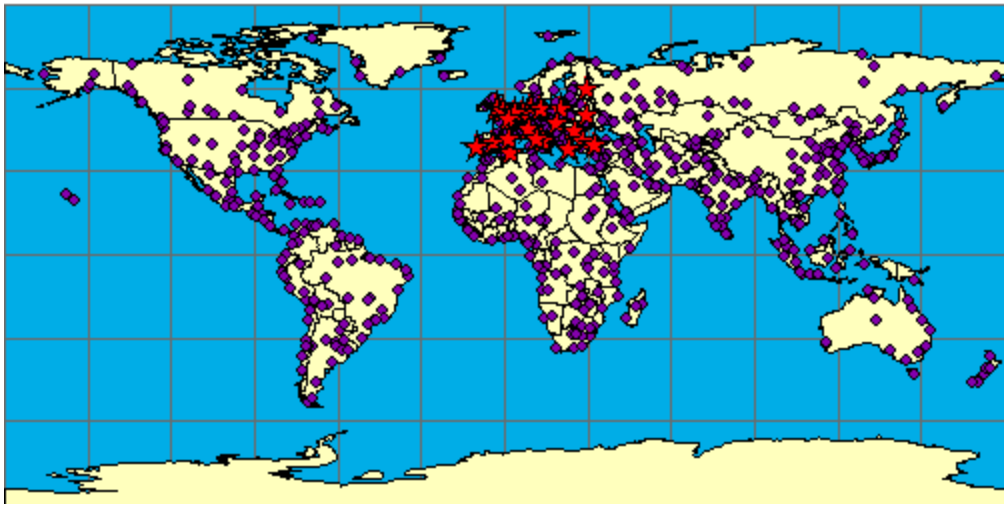
A spatial filter can also be set on a layer using SPATIALFILTER in a SPATIALQUERY. In the next example, a spatial filter is set to include only cities in Europe. ENVELOPE is used to set the filter boundary, but polygons, lines, points, and buffers could also be used.

GET_IMAGE using an attribute query and spatial filter in LAYER:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <IMAGESIZE width="500" height="400" />
        <LAYERLIST>
          <LAYERDEF id="4" visible="true" />
          <LAYERDEF id="3" visible="true" />
          <LAYERDEF id="2" visible="false" />
          <LAYERDEF id="1" visible="false" />
```

```
          <LAYERDEF id="0" visible="true" />
        </LAYERLIST>
      </PROPERTIES>
      <LAYER type="featureclass" name="Cities" visible="true" id="10">
        <DATASET fromlayer="0" />
        <SPATIALQUERY where="POPULATION &gt; 2000000" >
          <SPATIALFILTER relation="area_intersection">
            <ENVELOPE minx="-14" miny="35" maxx="33" maxy="64" />
          </SPATIALFILTER>
        </SPATIALQUERY>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

In the returned image, only cities with a population greater than two million within Europe are rendered with the selection symbol defined in the ArcMap document.



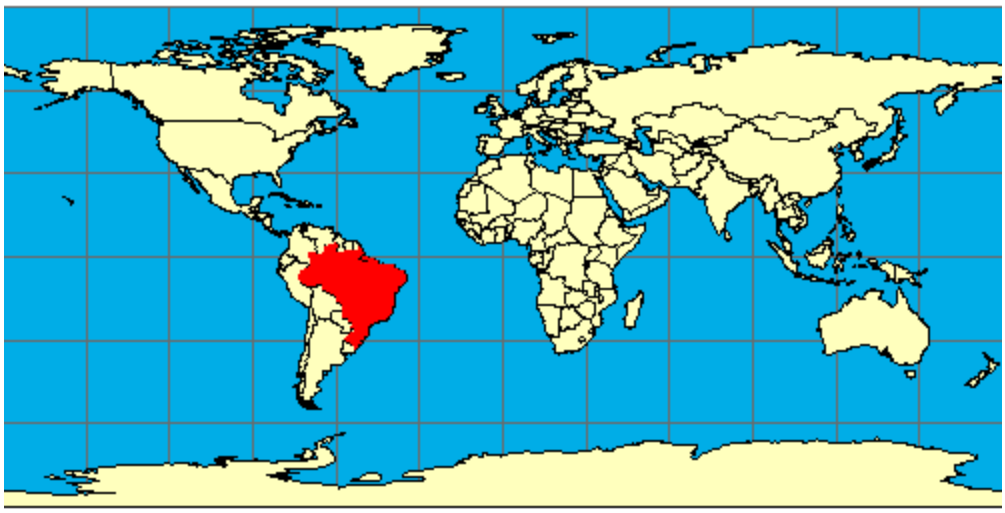
Method 2: Including a query in LAYERDEF. Attribute and spatial queries can be used in LAYERDEF instead of LAYER to produce the same results. With Image Services, a query in LAYERDEF will display only features in the filter. For example, assume a user selects Brazil from the Countries layer (*id="3"*). In an Image Service, only Brazil would be displayed. However, with ArcMap Image Services, when a query is used in LAYERDEF, all countries will be displayed, and Brazil is highlighted.

Using LAYERDEF to select Brazil:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <LAYERLIST>
          <LAYERDEF id="4" visible="true" /> <!--Ocean-->
          <LAYERDEF id="3" visible="true" > <!--Countries-->
            <SPATIALQUERY where="NAME=&apos;Brazil&apos;" />
```

```
          </LAYERDEF>
          <LAYERDEF id="2" visible="false" /> <!--States-->
          <LAYERDEF id="1" visible="false" /> <!--Provinces-->
          <LAYERDEF id="0" visible="false" /> <!--Cities-->
        </LAYERLIST>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

In the returned image, all countries are included and Brazil is highlighted based on the selection symbol defined in the ArcMap document.



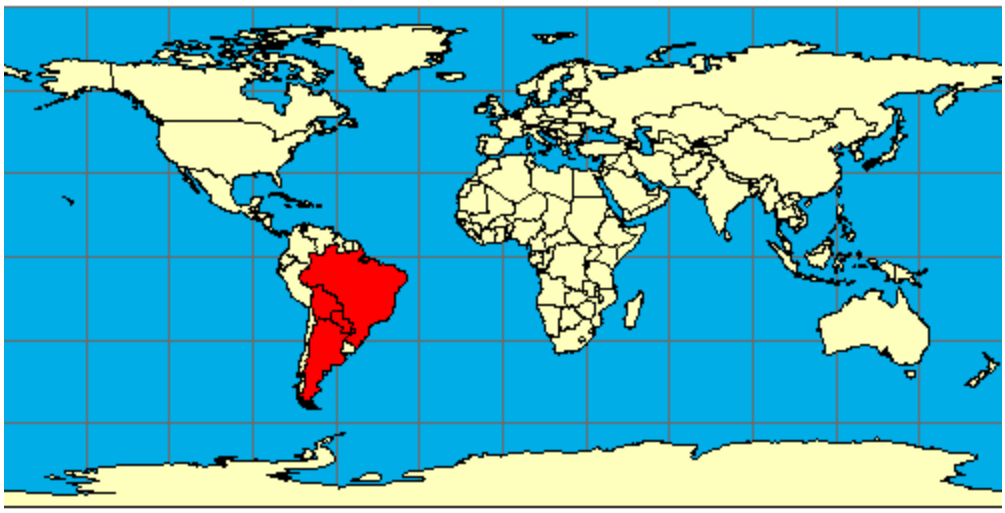
SPATIALFILTER can also be included in LAYERDEF. The next example uses an ENVELOPE as a spatial filter to select several countries in South America.

Using a spatial filter in LAYERDEF:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <LAYERLIST>
          <LAYERDEF id="4" visible="true" /> <!--Ocean-->
          <LAYERDEF id="3" visible="true" > <!--Countries-->
            <SPATIALQUERY>
              <SPATIALFILTER relation="area_intersection">
                <ENVELOPE minx="-63.6" miny="-26.9" maxx="-50.1"
maxy="-8.6" />
              </SPATIALFILTER>
            </SPATIALQUERY>
          </LAYERDEF>
```

```
          <LAYERDEF id="2" visible="false" /> <!--States-->
          <LAYERDEF id="1" visible="false" /> <!--Provinces-->
          <LAYERDEF id="0" visible="false" /> <!--Cities-->
        </LAYERLIST>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Once again, all features from the layer are included and features that are within the filter are highlighted with the selection symbol defined in the ArcMap document.



Adding acetate layers

Acetate layers display additional features on top of a map. The acetate layers are visible only in the HTML Viewer and viewers using the ColdFusion, ActiveX, or Java Connectors. They are not visible in ArcExplorer 4 or the Java Viewers.

Acetate layers are made up of one or more OBJECTs. The different objects include points, lines, polygons, text, north arrows, and scale bars. For details on the OBJECT element and its child elements, refer to the Notes section of OBJECT. Each object is placed on the acetate layer using screen coordinates or database coordinates. Screen coordinates are in pixels referenced from the lower left corner of the map frame. Database coordinates are in the coordinate system of the map.

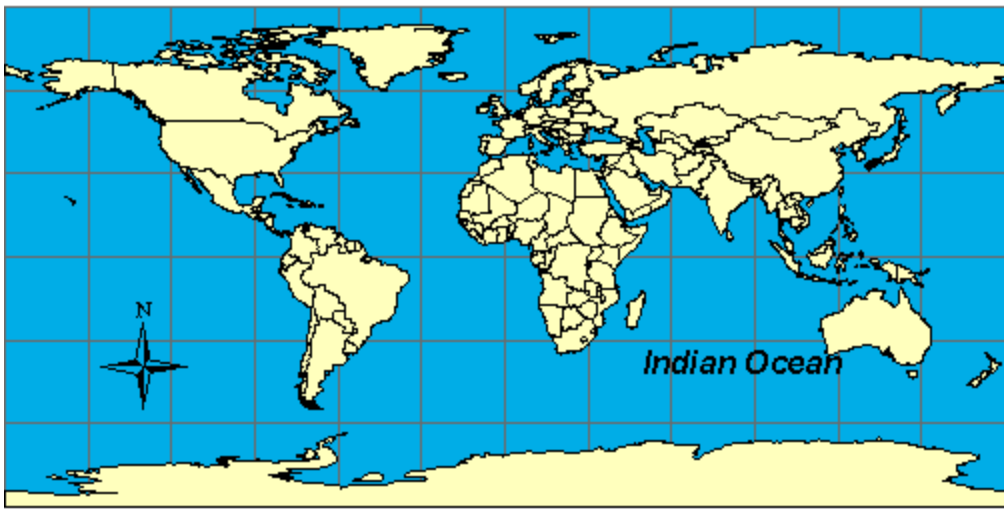
In the example below, a north arrow is added using screen coordinates. "Indian Ocean" is added as text using map coordinates.

Using a dynamic LAYER to add an acetate layer:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
```

```
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      <IMAGESIZE width="500" height="400" />
        <LAYERLIST>
          <LAYERDEF id="4" visible="true" />
          <LAYERDEF id="3" visible="true" />
          <LAYERDEF id="2" visible="false" />
          <LAYERDEF id="1" visible="false" />
          <LAYERDEF id="0" visible="false" />
        </LAYERLIST>
      </PROPERTIES>
      <LAYER type="acetate" name="acetate" id="acetate">
        <OBJECT units="database">
          <TEXT coords="50 -45" label="Indian Ocean">
            <TEXTMARKERSYMBOL fontstyle="bolditalic" fontsize="12" />
          </TEXT>
        </OBJECT>
        <OBJECT units="pixel">
          <NORTHARROW type="1" size="50" coords="70 150" />
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The returned map includes the two objects from the acetate layer: the north arrow and text.



Layer order and acetate layers

Acetate layers are always drawn on top of the service layers in the same order in which they are listed in the request. When LAYERLIST *order* is set to "true", the acetate layers must be included in the LAYERLIST or they will not be displayed. The next example includes the Ocean and Countries layers along with north arrow and text acetate layers. The acetate layer information is included in both the LAYER section and in the layer list.

Including acetate layers in the layer list when LAYERLIST order is set to "true":

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      <IMAGESIZE width="500" height="400" />
        <LAYERLIST order="true" >
          <LAYERDEF id="4" />
          <LAYERDEF id="3" />
          <LAYERDEF id="text" />
          <LAYERDEF id="northarrow" />
        </LAYERLIST>
      </PROPERTIES>
      <LAYER type="acetate" name="text" id="text">
        <OBJECT units="database">
          <TEXT coords="50 -45" label="Indian Ocean">
            <TEXTMARKERSYMBOL fontstyle="bolditalic" fontsize="12" />
          </TEXT>
        </OBJECT>
      </LAYER>
      <LAYER type="acetate" name="northarrow" id="northarrow">
        <OBJECT units="pixel">
          <NORTHARROW type="1" size="50" coords="70 150" />
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Using Projections with GET_IMAGE

The projection of an image can be changed using the projection elements:

- COORDSYS
- FEATURECOORDSYS
- FILTERCOORDSYS

For a complete discussion on the different elements, refer to Using Projection Elements.

FILTERCOORDSYS is used to specify the coordinate system of the requesting client. All the examples so far have been in geographic coordinates (decimal degrees), which have an ID of "4326".

FEATURECOORDSYS is used to specify to which coordinate system the service should be transformed. In the next example, the image is requested in Robinson, which has an ID of "54030".

Using FILTERCOORDSYS and FEATURECOORDSYS:

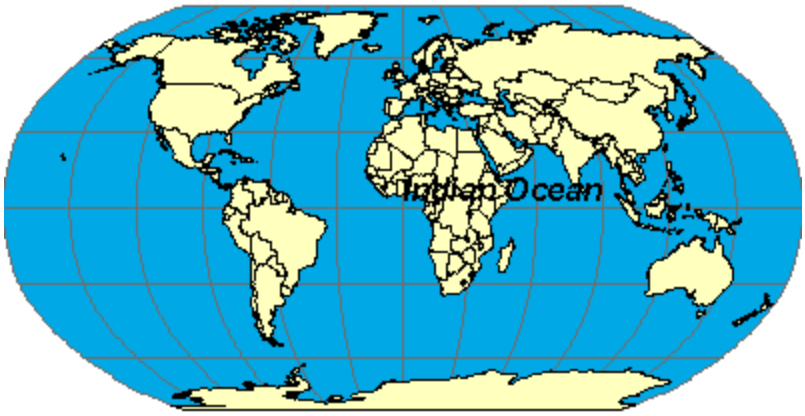
```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <FILTERCOORDSYS id="4326" />
        <FEATURECOORDSYS id="54030" />
        <LAYERLIST>
          <LAYERDEF id="4" visible="true" />
          <LAYERDEF id="3" visible="true" />
          <LAYERDEF id="2" visible="false" />
          <LAYERDEF id="1" visible="false" />
          <LAYERDEF id="0" visible="false" />
        </LAYERLIST>
      </PROPERTIES>
      <LAYER type="acetate" name="acetate" id="acetate">
        <OBJECT units="database">
          <TEXT coords="50 -45" label="Indian Ocean">
            <TEXTMARKERSYMBOL fontstyle="bolditalic" fontsize="12" />
          </TEXT>
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

In the response, the ENVELOPE coordinates are in Robinson, which was the coordinate system used in FEATURECOORDSYS in the request.

IMAGE response with new ENVELOPE coordinates:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-16986708.7102836" miny="-12740031.5327127"
maxx="16986708.7102836" maxy="12740031.5327127" />
      <OUTPUT
url="http://mycomputer.domain.com/output/world_MYCOMPUTER29852440.png"
file="C:\ArcIMS\output\world_MYCOMPUTER29852440.png" />
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

The returned map image shows the map in the Robinson coordinate system.

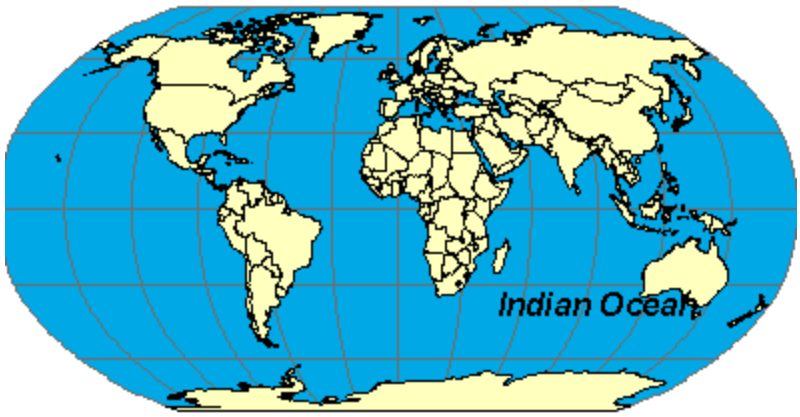


Note that the acetate OBJECT with the text "Indian Ocean" is no longer placed correctly. The reason is the OBJECT coordinate units are in geographic coordinates, while the map has been transformed to Robinson. COORDSYS can be used with an acetate OBJECT to let the ArcIMS Spatial Server know the coordinate system of the OBJECT. In the example below, COORDSYS has been added to the "Indian Ocean" object.

Including COORDSYS in the acetate layer:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <FILTERCOORDSYS id="4326" />
        <FEATURECOORDSYS id="54030" />
        <LAYERLIST>
          <LAYERDEF id="4" visible="true" />
          <LAYERDEF id="3" visible="true" />
          <LAYERDEF id="2" visible="false" />
          <LAYERDEF id="1" visible="false" />
          <LAYERDEF id="0" visible="false" />
        </LAYERLIST>
      </PROPERTIES>
      <LAYER type="acetate" name="acetate" id="acetate">
        <OBJECT units="database">
          <COORDSYS id="4326"/>
          <TEXT coords="50 -45" label="Indian Ocean">
            <TEXTMARKERSYMBOL fontstyle="bolditalic" fontsize="12" />
          </TEXT>
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

The returned image shows the map and the "Indian Ocean" OBJECT in the Robinson coordinate system.



Using LEGEND and DRAW

LEGEND can be used in a GET_IMAGE request to include a legend for the requested map. The legend is an additional image in the same format as the map: JPG, PNG8, or PNG24. DRAW can be used to request only a legend rather than both a legend and a map.

When using LEGEND with the ArcMap Server, the attribute *autoextend* should always be included and set to "true". The following GET_IMAGE request includes a default LEGEND using only the attribute *autoextend*. DRAW has the attribute *map* set to "false" so that only a legend is returned.

GET_IMAGE request with LEGEND:

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

In the response, only the name and location of the legend are included.

IMAGE response with LEGEND:

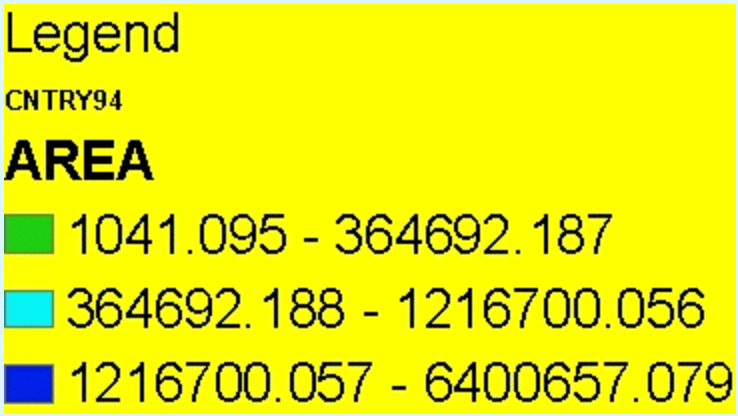
```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <LEGEND
```

```
url="http://mycomputer.domain.com/output/world_MYCOMPUTER1248849.png"
file="c:\arcims\output\world_MYCOMPUTER1248849.png" />
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

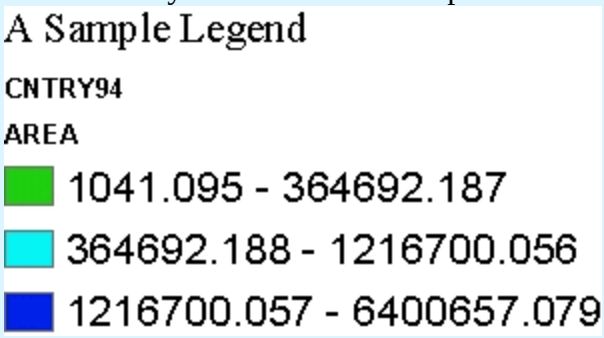


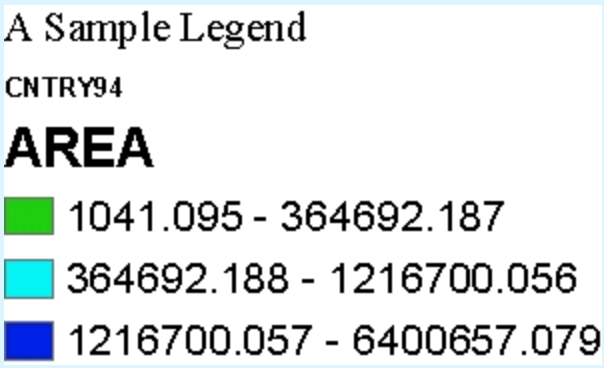
The returned legend image includes all layers that are currently visible in the map on a white background.

A limitation with ArcMap Server is the height and width of a legend cannot be controlled. The height and width are only big enough to accommodate information for all layers. Text does not wrap. A legend can get quite wide as the amount of text describing a layer increases or the font size increases.



Several attributes can be used with LEGEND to change the look and feel of the legend image. The following table shows the results of different attribute combinations.

	When using ***backgroundcolor***. The background color can be changed using RGB values. <LEGEND autoextend="true" backgroundcolor="255,255,0" />
	When using ***title, font,*** and ***titlefontsize***. A title can be added to the legend. The font and font size can also be included. The *font* attribute affects only text in the title and is not used with the layers. <LEGEND autoextend="true" title="A Sample Legend" font="Times New Roman" titlefontsize="16" />
Layers do not include a value map:  Layer includes a value map: 	When using ***valuefontsize***. For layers that include a value map, *valuefontsize* controls the font size of the labels that make up the value map. For layers with no value map, it controls the font size for the layer name. The font style and color cannot be changed. <LEGEND autoextend="true" title="A Sample Legend" font="Times New Roman" titlefontsize="16" valuefontsize="16" />

	When using ***layerfontsize***. Sets the font size of the subheading for layers that include a value map. The font style and color cannot be changed. In this example, the subheading is "AREA". <LEGEND autoextend="true" title="A Sample Legend" font="Times New Roman" titlefontsize="16" valuefontsize="16" layerfontsize="20" />
	When using ***swatchheight*** and ***swatchwidth***. The size of the swatch representing each layer can be increased or decreased using *swatchheight* and *swatchwidth*. Measurements are in pixels. <LEGEND autoextend="true" swatchheight="25" swatchwidth="25" />

Using OUTPUT to Control Image Names and Locations

When a service is started in ArcIMS Administrator, the image output directory and URL location are determined at that time. When a request is made, the ArcIMS Spatial Server assigns a filename. The name includes the service name, the computer name on which the image was generated, and a randomly generated number. If the service is named "world" and the computer is "MYCOMPUTER", then an example filename is "world_MYCOMPUTER1248849.png". The ArcIMS Tasker Windows service or UNIX daemon automatically deletes these generated images on a user-specified interval.

When OUTPUT is used in a GET_IMAGE request, it can be used to override location information from when the service was started. An important note is that the output files are not automatically deleted by ArcIMS Tasker. In order for the files to be deleted, the *taskfile* property must be set in tasker.properties. For information on setting this property, see the ArcIMS Help.

OUTPUT works with paired attributes. If one of the attributes is used, its pair is required. There are attribute pairs for both the map and legend images. Determining which pair to use depends on whether the ArcIMS Spatial Server defines the output filename or the user does. In all cases, the output directory and URL are user-specified.

Attribute	Paired Attribute	Filename assignment	Example: http://mycomputer/arcims/...
path	baseurl	ArcIMS assigns random filename.	world_MYCOMPUTER1248849.png
name	url	User assigns a filename.	myfilename.png
legendpath	legendbaseurl	ArcIMS assigns random filename.	world_MYCOMPUTER1248851.png
legendname	legendurl	User assigns a filename.	myfilename.png

When starting a service or using OUTPUT in a request, UNC pathnames are valid. For example, instead of using "c:\arcims\output", "\\myComputer\arcims\output" can be used.

ArcIMS Spatial Server assigns filename

In the following example, the ArcIMS Spatial Server assigns a filename, but the user assigns the directory and URL for both the map and the legend. The map and the legend images do not need to be written to the same directory. The attribute pairs in this scenario are *path-baseurl* and *legendpath-legendbaseurl*.

Note that even though *legendpath* and *legendbaseurl* are included in OUTPUT, the LEGEND element *must* also be present to generate a legend image. If LEGEND is not included, no legend is generated.

OUTPUT when ArcIMS Spatial Server defines filename:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <OUTPUT path="c:\arcims"
baseurl="http://mycomputer.domain.com/arcims"
legendpath="c:\arcims\legend"
legendbaseurl="http://mycomputer.domain.com/arcims/legend" />
        <LEGEND />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

IMAGE response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
```

```
    <IMAGE>
      <ENVELOPE minx="-180" miny="-144" maxx="180" maxy="144" />
      <OUTPUT
url="http://mycomputer.domain.com/world_MYCOMPUTER2983738.png"
file="c:\arcims\world_MYCOMPUTER2983738.png" />
      <LEGEND
url="http://mycomputer.domain.com/output/world_MYCOMPUTER2983739.png"
file="c:\arcims\legend\world_MYCOMPUTER2983739.png" />
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

User assigns filename

In the next example, the user assigns a filename, directory, and URL for both the map and the legend. The attribute pairs in this scenario are *name-url* and *legendname-legendurl*. Once again, to generate a legend, LEGEND must be included in either the map configuration file or the request.

OUTPUT when user defines filename:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <OUTPUT name="c:\arcims\mapimage.png"
url="http://mycomputer.domain.com/arcims/mapimage.png"
legendname="c:\arcims\legend\legendimage.png"
legendurl="http://mycomputer.domain.com/arcims/legend/legendimage.png"
/>
        <LEGEND />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

IMAGE response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-180" miny="-144" maxx="180" maxy="144" />
      <OUTPUT url="http://mycomputer.domain.com/mapimage.png"
file="c:\arcims\mapimage.png" />
      <LEGEND url="http://mycomputer.domain.com/legend/legendimage.png"
file="c:\arcims\legend\legendimage.png" />
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

Restricting OUTPUT

When using an ArcIMS HTML Viewer, ArcIMS Java Viewer, ArcExplorer 4, or any other client using the ArcIMS Servlet Connector, the OUTPUT element is restricted by default.

- In a GET_IMAGE request, OUTPUT is ignored, even if it is included in the request.

GET_IMAGE request when OUTPUT is restricted:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <OUTPUT name="c:\arcims\mapimage.png"
url="http://mycomputer.domain.com/arcims/mapimage.png" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

- In an IMAGE response, the attribute *file* in OUTPUT is not returned; only *url* is included. The OUTPUT location is the default location specified when the service was started. The ArcIMS Spatial Server assigns the filename.

IMAGE response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-180" miny="-144" maxx="180" maxy="144" />
      <OUTPUT
url="http://mycomputer.domain.com/output/world_MYCOMPUTER2102209.png"
/>
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

These restrictions can be lifted by setting the properties *spatialServer.AllowRequestOutput* and *spatialServer.AllowResponsePath* to true in Esrimap_prop. This property file is found in the servlet engine directory. For more information on the location of Esrimap_prop and its properties, see the ArcIMS Help.

These restrictions apply only when the ArcIMS Servlet Connector is used. They do not apply to the ActiveX, ColdFusion, or Java Connectors. The OUTPUT filename extensions are restricted to *.jpg and *.png, regardless of whether OUTPUT is restricted or not.

Comparison of Elements and Attributes used with the Image and ArcMap Servers

This document compares elements, child elements and attributes used for the following requests and responses:

- GET_IMAGE and IMAGE
- GET_FEATURES and FEATURES
- GET_SERVICE_INFO and SERVICEINFO
- GET_LAYOUT and LAYOUT

ARCXML	Image - YES	ArcMap - YES
Attributes:		
version	Yes	Yes
Child Elements:		
REQUEST	Yes	Yes
RESPONSE	Yes	Yes

BACKGROUND	Image - YES	ArcMap - YES
Attributes:		
color	Yes	Yes
transcolor	Yes	Yes
No Child Elements		

BAND	Image - YES	ArcMap - YES
Attributes:		
number	Yes	Yes
value	Yes	Yes
No Child Elements		

BANDS	Image - YES	ArcMap - YES
Attributes:		

rasterid	Yes	Yes

Child Elements:

BAND	Yes	Yes

BUFFER	**Image - YES**	**ArcMap - YES**
Attributes:		
bufferunits	Yes	Yes
distance	Yes	Yes
project	Yes	No

Child Elements:

*When parent element is **SPATIALFILTER:***
No Child Elements

*When parent element is **SPATIALQUERY:***

SPATIALQUERY	Yes	Yes
TARGETLAYER	Yes	Yes

CALLOUTMARKERSYMBOL	**Image - YES**	**ArcMap - NO**
Attributes:		
antialiasing	Yes	No
backcolor	Yes	No
boundarycolor	Yes	No
font	Yes	No
fontcolor	Yes	No
fontsize	Yes	No
fontstyle	Yes	No
glowing	Yes	No
interval	Yes	No
outline	Yes	No
shadow	Yes	No
transparency	Yes	No

No Child Elements

CAPABILITIES	Image - YES	ArcMap - YES
Attributes:		
forbidden	Yes	Yes
No Child Elements		

CHARTSYMBOL	Image - YES	ArcMap - NO
Attributes:		
antialiasing	Yes	No
maxsize	Yes	No
maxvalue	Yes	No
minsize	Yes	No
minvalue	Yes	No
mode	Yes	No
outline	Yes	No
shadow	Yes	No
size	Yes	No
sizefield	Yes	No
transparency	Yes	No
Child Elements:		
CHARTVALUE	Yes	No

CHARTVALUE	Image - YES	ArcMap - NO
Attributes:		
color	Yes	No
lookupfield	Yes	No
lower	Yes	No
upper	Yes	No
value	Yes	No
No Child Elements		

COORDS	Image - YES	ArcMap - YES
No Attributes		
No Child Elements		

COORDSYS	Image - YES	ArcMap - YES

Attributes:		
datumtransformid	Yes	No
datumtransformstring	Yes	No
id	Yes	Yes
string	Yes	Yes
No Child Elements		

DATAFRAME	**Image - NO**	**ArcMap - YES**
Attributes:		
id	No	Yes
Child Elements:		
ENVELOPE	No	Yes
FEATURECOORDSYS	No	Yes
FILTERCOORDSYS	No	Yes

DATAFRAMEINFO	**Image - NO**	**ArcMap - YES**
No Attributes		
Child Elements:		
LAYERINFO	No	Yes
PROPERTIES	No	Yes

DATASET	**Image - YES**	**ArcMap - YES**
Attributes:		
*When parent element is **LAYER and the layer type is featureclass:***		
name	Yes	No
type	Yes	No
workspace	Yes	No
*When parent element is **LAYER and the layer type is image:***		
name	Yes	No
type	Yes	No

workspace	Yes	No

*When parent element is **LAYER in GET_IMAGE or GET_EXTRACT:***

fromlayer	Yes	Yes

No Child Elements

DENSIFY **Image - YES** **ArcMap - NO**

Attributes:

tolerance	Yes	No

No Child Elements

DRAW **Image - YES** **ArcMap - YES**

Attributes:

map	Yes	Yes

No Child Elements

ENVELOPE **Image - YES** **ArcMap - YES**

Attributes:

*When parent element is **ALL except PROPERTIES:***

maxx	Yes	Yes
maxy	Yes	Yes
minx	Yes	Yes
miny	Yes	Yes

*When parent element is **PROPERTIES:***

maxx	Yes	Yes
maxy	Yes	Yes
minx	Yes	Yes
miny	Yes	Yes
reaspect	Yes	No

No Child Elements

ENVIRONMENT **Image - YES** **ArcMap - YES**

No Attributes

Child Elements:

*When parent element is **GET_EXTRACT, GET_FEATURES, GET_IMAGE:***

SEPARATORS	Yes	Yes

*When parent element is **SERVICEINFO:***

CAPABILITIES	Yes	Yes
IMAGELIMIT	Yes	Yes
LOCALE	Yes	Yes
SCREEN	Yes	Yes
SEPARATORS	Yes	Yes
UIFONT	Yes	Yes

ERROR	**Image - YES**	**ArcMap - YES**
Attributes:		
machine	Yes	Yes
processid	Yes	Yes
threadid	Yes	Yes

No Child Elements

EXACT	**Image - YES**	**ArcMap - NO**
Attributes:		
label	Yes	No
method	Yes	No
value	Yes	No

Child Elements:

*When parent element is **VALUEMAPLABELRENDERER:***

CALLOUTMARKERSYMBOL	Yes	No
CHARTSYMBOL	Yes	No

RASTERSHIELDSYMBOL	Yes	No
SHIELDSYMBOL	Yes	No
TEXTSYMBOL	Yes	No
*When parent element is **VALUEMAPRENDERER:***		
GRADIENTFILLSYMBOL	Yes	No
HASHLINESYMBOL	Yes	No
RASTERFILLSYMBOL	Yes	No
RASTERMARKERSYMBOL	Yes	No
SIMPLELINESYMBOL	Yes	No
SIMPLEMARKERSYMBOL	Yes	No
SIMPLEPOLYGONSYMBOL	Yes	No
TRUETYPEMARKERSYMBOL	Yes	No

EXTENSION	**Image - YES**	**ArcMap - NO**
Attributes:		
type	Yes	No
Child Elements:		
EXTRACTPARAMS	Yes	No
GCSTYLE	Yes	No
STOREDQUERIES	Yes	No

FCLASS	**Image - YES**	**ArcMap - YES**
Attributes:		
type	Yes	Yes

Child Elements:

When using all servers except ArcMap Server:

ENVELOPE	Yes	No
FIELD	Yes	No

When using ArcMap Server:

ENVELOPE	No	Yes
FIELD	No	Yes
TOC	No	Yes

FEATURE — Image - YES — ArcMap - YES

No Attributes

Child Elements:

*When parent element is **FEATURES:***

ENVELOPE	Yes	Yes
FIELDS	Yes	Yes
MULTIPOINT	Yes	Yes
POLYGON	Yes	Yes
POLYLINE	Yes	Yes

FEATURECOORDSYS — Image - YES — ArcMap - YES

Attributes:

datumtransformid	Yes	No
datumtransformstring	Yes	No
id	Yes	Yes
string	Yes	Yes

No Child Elements

FEATURECOUNT	Image - YES	ArcMap - YES
Attributes:		
count	Yes	Yes
hasmore	Yes	Yes
No Child Elements		

FEATURES	Image - YES	ArcMap - YES
No Attributes		
Child Elements:		
ENVELOPE	Yes	Yes
FEATURE	Yes	Yes
FEATURECOUNT	Yes	Yes

FIELD	Image - YES	ArcMap - YES
Attributes:		
*When parent element is **FIELDS:***		
name	Yes	Yes
value	Yes	Yes
Child Elements:		
*When parent element is **SQVAR, FIELDS, FCLASS:***		
No Child Elements		

FIELDS	Image - YES	ArcMap - YES
No Attributes		
Child Elements:		
FIELD	Yes	Yes

FILTERCOORDSYS	Image - YES	ArcMap - YES
Attributes:		

datumtransformid	Yes	No
datumtransformstring	Yes	No
id	Yes	Yes
string	Yes	Yes

No Child Elements

GET_FEATURES	**Image - YES**	**ArcMap - YES**
Attributes:		
attributes	Yes	Yes
beginrecord	Yes	Yes
checkesc	Yes	Yes
compact	Yes	Yes
dataframe	No	Yes
envelope	Yes	Yes
featurelimit	Yes	Yes
geometry	Yes	Yes
globalenvelope	Yes	Yes
outputmode	Yes	Yes
skipfeatures	Yes	Yes
Child Elements:		
ENVIRONMENT	Yes	Yes
LAYER	Yes	Yes
QUERY	Yes	Yes
SPATIALQUERY	Yes	Yes

GET_IMAGE	**Image - YES**	**ArcMap - YES**
Attributes:		
dataframe	No	Yes
autoresize	Yes	Yes
show	Yes	No
Child Elements:		
ENVIRONMENT	Yes	Yes

LAYER	Yes	Yes
PROPERTIES	Yes	Yes
WORKSPACES	Yes	No

GET_LAYOUT	**Image - NO**	**ArcMap - YES**
Attributes:		
autoresize	No	Yes
Child Elements:		
DATAFRAME	No	Yes
PROPERTIES	No	Yes

GET_RASTER_INFO	**Image - YES**	**ArcMap - YES**
Attributes:		
dataframe	No	Yes
layerid	Yes	Yes
x	Yes	Yes
y	Yes	Yes
Child Elements:		
COORDSYS	Yes	Yes

GET_SERVICE_INFO	**Image - YES**	**ArcMap - YES**
Attributes:		
*When parent element is **REQUEST for an ArcMap Image Service:***		
dataframe	No	Yes
envelope	No	Yes
fields	No	Yes
toc	No	Yes
toctype	No	Yes
*When parent element is **REQUEST for an Image or Feature Service:***		
dpi	Yes	No

envelope	Yes	No
extensions	Yes	No
fields	Yes	No
renderer	Yes	No

No Child Elements

GRADIENTFILLSYMBOL	**Image - YES**	**ArcMap - YES**	
Attributes:			
antialiasing	Yes	No	
finishcolor	Yes	Yes	Acetate layer only in ArcMap Server.
overlap	Yes	No	
startcolor	Yes	Yes	Acetate layer only in ArcMap Server.
transparency	Yes	No	
type	Yes	Yes	Acetate layer only in ArcMap Server.

No Child Elements

GROUPRENDERER	**Image - YES**	**ArcMap - NO**
No Attributes		
Child Elements:		
GROUPRENDERER	Yes	No
SCALEDEPENDENTRENDERER	Yes	No
SIMPLELABELRENDERER	Yes	No
SIMPLERENDERER	Yes	No
VALUEMAPLABELRENDERER	Yes	No

VALUEMAPRENDERER	Yes	No	

HASHLINESYMBOL	**Image - YES**	**ArcMap - YES**	
Attributes:			
antialiasing	Yes	No	
color	Yes	Yes	Acetate layer only in ArcMap Server.
interval	Yes	Yes	Acetate layer only in ArcMap Server.
linethickness	Yes	Yes	Acetate layer only in ArcMap Server.
overlap	Yes	No	
tickthickness	Yes	Yes	Acetate layer only in ArcMap Server.
transparency	Yes	No	
type	Yes	Yes	Acetate layer only in ArcMap Server.
width	Yes	Yes	Acetate layer only in ArcMap Server.

No Child Elements

HOLE	**Image - YES**	**ArcMap - YES**
No Attributes		
Child Elements:		
COORDS	Yes	Yes
POINT	Yes	Yes

IMAGE	**Image - YES**	**ArcMap - YES**
No Attributes		
Child Elements:		
ENVELOPE	Yes	Yes
LAYERS	Yes	No

LEGEND	Yes	Yes
OUTPUT	Yes	Yes

IMAGEGENERALIZATION **Image - YES** **ArcMap - NO**

No Attributes

No Child Elements

IMAGELIMIT **Image - YES** **ArcMap - YES**

Attributes:

pixelcount	Yes	Yes

No Child Elements

IMAGEPROPERTIES **Image - YES** **ArcMap - NO**

Attributes:

transcolor	Yes	No
transparency	Yes	No

No Child Elements

IMAGESIZE **Image - YES** **ArcMap - YES**

Attributes:

*When parent element is **GET_EXTRACT, GET_LAYOUT:***

height	No	Yes
width	No	Yes

*When parent element is **GET_IMAGE:***

dpi	Yes	Yes
height	Yes	Yes
printheight	Yes	No
printwidth	Yes	No
scalesymbols	Yes	No
width	Yes	Yes

No Child Elements

IMAGEWORKSPACE	Image - YES	ArcMap - NO
Attributes:		
directory	Yes	No
name	Yes	No
No Child Elements		

LAYER	Image - YES	ArcMap - YES
Attributes:		
*When parent element is **GET_FEATURES, GET_GEOCODE:***		
id	Yes	Yes
*When parent element is **LAYERS in a GET_IMAGE request or map configuration file:***		
id	Yes	No
*When parent element is **LAYERS in an IMAGE response:***		
featurecount	Yes	No
id	Yes	No
name	Yes	No
*When parent element is **MAP, GET_IMAGE, GET_EXTRACT:***		
id	Yes	No
maxscale	Yes	No
minscale	Yes	No
name	Yes	No
type	Yes	No
visible	Yes	No
Child Elements:		
*When parent element is **GET_FEATURES, GET_GEOCODE:***		
No Child Elements		
*When parent element is **GET_IMAGE and LAYER type="featureclass" or "image" - ArcMap Server:***		
DATASET	No	Yes
QUERY	No	Yes

SPATIALQUERY	No	Yes

*When parent element is **LAYERS:***

No Child Elements

*When parent element is **MAP or GET_IMAGE and LAYER type="acetate" - Image and ArcMap Servers:***

OBJECT	Yes	Yes

*When parent element is **MAP or GET_IMAGE and LAYER type="image" - Image Server:***

COORDSYS	Yes	No
DATASET	Yes	No
IMAGEPROPERTIES	Yes	No

*When parent element is **MAP, GET_IMAGE, or GET_EXTRACT and LAYER type="featureclass" - Image and Extract Servers:***

COORDSYS	Yes	No
DATASET	Yes	No
DENSIFY	Yes	No
EXTENSION	Yes	No
GROUPRENDERER	Yes	No
QUERY	Yes	No
SCALEDEPENDENTRENDERER	Yes	No
SIMPLELABELRENDERER	Yes	No
SIMPLERENDERER	Yes	No
SPATIALQUERY	Yes	No
VALUEMAPLABELRENDERER	Yes	No

VALUEMAPRENDERER	Yes	No

LAYERDEF — Image - YES — ArcMap - YES

Attributes:

*When parent element is **LAYERLIST in GET_IMAGE or GET_EXTRACT request:***

id	Yes	Yes
name	Yes	Yes
visible	Yes	Yes

Child Elements:

*When parent element is **LAYERLIST in GET_IMAGE request - ArcMap Server:***

QUERY	No	Yes
SPATIALQUERY	No	Yes

*When parent element is **LAYERLIST in GET_IMAGE request - Image Server:***

GROUPRENDERER	Yes	No
QUERY	Yes	No
SCALEDEPENDENTRENDERER	Yes	No
SIMPLELABELRENDERER	Yes	No
SIMPLERENDERER	Yes	No
SPATIALQUERY	Yes	No
VALUEMAPLABELRENDERER	Yes	No
VALUEMAPRENDERER	Yes	No

LAYERINFO — Image - YES — ArcMap - YES

Attributes:

id	Yes	Yes
maxscale	Yes	Yes
minscale	Yes	Yes
name	Yes	Yes
type	Yes	Yes
visible	Yes	Yes

Child Elements:

*When parent element is **LAYERINFO and type is acetate:***

No Child Elements

*When parent element is **LAYERINFO and type is featureclass for ArcMap Image Services:***

FCLASS	No	Yes

*When parent element is **LAYERINFO and type is featureclass for Image or Feature Services:***

EXTENSION	Yes	No
FCLASS	Yes	No
GROUPRENDERER	Yes	No
SCALEDEPENDENTRENDERER	Yes	No
SIMPLELABELRENDERER	Yes	No
SIMPLERENDERER	Yes	No
VALUEMAPLABELRENDERER	Yes	No
VALUEMAPRENDERER	Yes	No

*When parent element is **LAYERINFO and type is image:***

ENVELOPE	Yes	Yes

LAYERLIST — Image - YES — ArcMap - YES

Attributes:

dynamicfirst	Yes	No

nodefault	Yes	No
order	Yes	Yes
Child Elements:		
LAYERDEF	Yes	Yes

LAYERS	**Image - YES**	**ArcMap - NO**
No Attributes		
Child Elements:		
LAYER	Yes	No

LAYOUT	**Image - NO**	**ArcMap - YES**
No Attributes		
Child Elements:		
ENVELOPE	No	Yes
OUTPUT	No	Yes

LAYOUTINFO	**Image - NO**	**ArcMap - YES**
Attributes:		
pageunits	No	Yes
Child Elements:		
ENVELOPE	No	Yes

LEGEND	**Image - YES**	**ArcMap - YES**
Attributes:		
*When parent element is **IMAGE:***		
file	Yes	Yes
url	Yes	Yes

*When parent element is **PROPERTIES:***

anitaliasing	Yes	No
autoextend	Yes	Yes
backgroundcolor	Yes	Yes
cansplit	Yes	No
cellspacing	Yes	No
columns	Yes	No
display	Yes	Yes
font	Yes	Yes
height	Yes	No
layerfontsize	Yes	Yes
reverseorder	Yes	No
splittext	Yes	No
swatchheight	Yes	Yes
swatchwidth	Yes	Yes
title	Yes	Yes
titlefontsize	Yes	Yes
transcolor	Yes	No
valuefontsize	Yes	Yes
width	Yes	No

Child Elements:

*When parent element is **IMAGE:***

No Child Elements

*When parent element is **PROPERTIES:***

LAYERS	Yes	No

LINE	**Image - YES**	**ArcMap - YES**
Attributes:		
coords	Yes	Yes
Child Elements:		
HASHLINESYMBOL	Yes	Yes
RASTERMARKERSYMBOL	Yes	Yes
SIMPLELINESYMBOL	Yes	Yes

SIMPLEMARKERSYMBOL	Yes	Yes
TRUETYPEMARKERSYMBOL	Yes	Yes

LOCALE	**Image - YES**	**ArcMap - YES**
Attributes:		
country	Yes	Yes
language	Yes	Yes
variant	Yes	Yes
No Child Elements		

MAPUNITS	**Image - YES**	**ArcMap - YES**
Attributes:		
units	Yes	Yes
No Child Elements		

MULTIPOINT	**Image - YES**	**ArcMap - YES**
No Attributes		
Child Elements:		
COORDS	Yes	Yes
POINT	Yes	Yes

NORTHARROW	**Image - YES**	**ArcMap - YES**
Attributes:		
angle	Yes	Yes
antialiasing	Yes	No
coords	Yes	Yes
outline	Yes	No
overlap	Yes	No
shadow	Yes	No
size	Yes	Yes
transparency	Yes	No
type	Yes	Yes

No Child Elements

OBJECT	Image - YES	ArcMap - YES
Attributes:		
lower	Yes	No
units	Yes	Yes
upper	Yes	No
Child Elements:		
***For lines, option 1 with Image and ArcMap Image Services is to use* LINE:**		
COORDSYS	Yes	Yes
LINE	Yes	Yes
***For lines, option 2 with ArcMap Image Services is to use* POLYLINE *and one symbol*:**		
COORDSYS	Yes	Yes
HASHLINESYMBOL	Yes	Yes
POLYLINE	Yes	Yes
SIMPLELINESYMBOL	Yes	Yes
***For lines, option 2 with Image Services is to use* POLYLINE *and one symbol*:**		
COORDSYS	Yes	Yes
HASHLINESYMBOL	Yes	Yes
POLYLINE	Yes	Yes
RASTERMARKERSYMBOL	Yes	Yes
SIMPLELINESYMBOL	Yes	Yes
SIMPLEMARKERSYMBOL	Yes	Yes
TRUETYPEMARKERSYMBOL	Yes	Yes

***For points, option 1 with Image and ArcMap Image Services is to use* POINT:**		
COORDSYS	Yes	Yes
POINT	Yes	Yes
***For points, option 2 with Image and ArcMap Image Services is to use* MULTIPOINT *and one symbol*:**		
COORDSYS	Yes	Yes
MULTIPOINT	Yes	Yes
RASTERMARKERSYMBOL	Yes	Yes
SIMPLEMARKERSYMBOL	Yes	Yes
TRUETYPEMARKERSYMBOL	Yes	Yes
***For polygons, option 1 with Image and ArcMap Image Services is to use* POLYGON. RING and HOLE are not permitted:**		
COORDSYS	Yes	Yes
POLYGON	Yes	Yes
***For polygons, option 2 with ArcMap Image Services is to use* POLYGON *and one symbol*. RING and HOLE are permitted:**		
COORDSYS	Yes	Yes
GRADIENTFILLSYMBOL	Yes	Yes
POLYGON	Yes	Yes
RASTERFILLSYMBOL	Yes	Yes
SIMPLEPOLYGONSYMBOL	Yes	Yes
***For polygons, option 2 with Image Services is to use* POLYGON *and one symbol*. RING and HOLE are permitted:**		
COORDSYS	Yes	Yes
GRADIENTFILLSYMBOL	Yes	Yes
HASHLINESYMBOL	Yes	Yes

POLYGON	Yes	Yes
RASTERFILLSYMBOL	Yes	Yes
RASTERMARKERSYMBOL	Yes	Yes
SIMPLELINESYMBOL	Yes	Yes
SIMPLEMARKERSYMBOL	Yes	Yes
SIMPLEPOLYGONSYMBOL	Yes	Yes
TRUETYPEMARKERSYMBOL	Yes	Yes

Other OBJECT child elements with Image and ArcMap Image Services:

COORDSYS	Yes	Yes
NORTHARROW	Yes	Yes
SCALEBAR	Yes	Yes
TEXT	Yes	Yes

OTHER	**Image - YES**	**ArcMap - NO**
Attributes:		
label	Yes	No
Child Elements:		
*When parent element is **VALUEMAPLABELRENDERER:***		
CALLOUTMARKERSYMBOL	Yes	No
CHARTSYMBOL	Yes	No
RASTERSHIELDSYMBOL	Yes	No
SHIELDSYMBOL	Yes	No

TEXTSYMBOL	Yes	No

*When parent element is **VALUEMAPRENDERER:***

GRADIENTFILLSYMBOL	Yes	No
HASHLINESYMBOL	Yes	No
RASTERFILLSYMBOL	Yes	No
RASTERMARKERSYMBOL	Yes	No
SIMPLELINESYMBOL	Yes	No
SIMPLEMARKERSYMBOL	Yes	No
SIMPLEPOLYGONSYMBOL	Yes	No
TRUETYPEMARKERSYMBOL	Yes	No

OUTPUT **Image - YES** **ArcMap - YES**

Attributes:

*When parent element is **IMAGE, EXTRACT, LAYOUT:***

file	Yes	Yes
height	Yes	Yes
url	Yes	Yes
width	Yes	Yes

*When parent element is **PROPERTIES in GET_IMAGE:***

baseurl	Yes	Yes
legendbaseurl	Yes	Yes
legendname	Yes	Yes
legendpath	Yes	Yes
legendurl	Yes	Yes
name	Yes	Yes
path	Yes	Yes
type	Yes	Yes
url	Yes	Yes

*When parent element is **PROPERTIES in GET_LAYOUT:***

baseurl	No	Yes
name	No	Yes
path	No	Yes
type	No	Yes
url	No	Yes

No Child Elements

PATH	**Image - YES**	**ArcMap - YES**

No Attributes

Child Elements:

COORDS	Yes	Yes
POINT	Yes	Yes

POINT	**Image - YES**	**ArcMap - YES**

Attributes:

*When parent element is **FIELDVALUE, MULTIPOINT, PATH, RING, HOLE:***

x	Yes	Yes
y	Yes	Yes

*When parent element is **OBJECT:***

coords	Yes	Yes

Child Elements:

*When parent element is **OBJECT:***

RASTERMARKERSYMBOL	Yes	Yes
SIMPLEMARKERSYMBOL	Yes	Yes
TRUETYPEMARKERSYMBOL	Yes	Yes

POLYGON	**Image - YES**	**ArcMap - YES**

Attributes:

*When parent element is **OBJECT:***

coords	Yes	Yes

Child Elements:

*When parent element is **FIELDVALUE, FEATURE, SPATIALFILTER, OBJECT:***

RING	Yes	Yes

*When parent element is **OBJECT:***

GRADIENTFILLSYMBOL	Yes	Yes
HASHLINESYMBOL	Yes	Yes
RASTERFILLSYMBOL	Yes	Yes
RASTERMARKERSYMBOL	Yes	Yes
SIMPLELINESYMBOL	Yes	Yes
SIMPLEMARKERSYMBOL	Yes	Yes
SIMPLEPOLYGONSYMBOL	Yes	Yes
TRUETYPEMARKERSYMBOL	Yes	Yes

POLYLINE — Image - YES — ArcMap - YES

No Attributes

Child Elements:

PATH	Yes	Yes

PROPERTIES — Image - YES — ArcMap - YES

No Attributes

Child Elements:

*When parent element is **DATAFRAMEINFO:***

ENVELOPE	No	Yes

FEATURECOORDSYS	No	Yes
FILTERCOORDSYS	No	Yes
MAPUNITS	No	Yes
*When parent element is **GET_IMAGE:***		
BACKGROUND	Yes	Yes
DRAW	Yes	Yes
ENVELOPE	Yes	Yes
FEATURECOORDSYS	Yes	Yes
FILTERCOORDSYS	Yes	Yes
IMAGESIZE	Yes	Yes
LAYERLIST	Yes	Yes
LEGEND	Yes	Yes
OUTPUT	Yes	Yes
*When parent element is **GET_LAYOUT:***		
ENVELOPE	No	Yes
FEATURECOORDSYS	No	Yes
FILTERCOORDSYS	No	Yes
IMAGESIZE	No	Yes
OUTPUT	No	Yes
*When parent element is **SERVICEINFO:***		
BACKGROUND	Yes	Yes
ENVELOPE	Yes	Yes

FEATURECOORDSYS	Yes	Yes
FILTERCOORDSYS	Yes	Yes
IMAGEGENERALIZATION	Yes	No
IMAGESIZE	Yes	No
LEGEND	Yes	No
MAPUNITS	Yes	Yes
OUTPUT	Yes	No

QUERY	**Image - YES**	**ArcMap - YES**
Attributes:		
accuracy	Yes	No
featurelimit	Yes	Yes
joinexpression	Yes	No
jointables	Yes	No
subfields	Yes	Yes
where	Yes	Yes
Child Elements:		
*When parent element is **GET_FEATURES:***		
BUFFER	Yes	Yes
FEATURECOORDSYS	Yes	Yes
*When parent element is **LAYER and LAYERDEF in GET_IMAGE:***		
BUFFER	Yes	Yes

RANGE	**Image - YES**	**ArcMap - NO**
Attributes:		
equality	Yes	No
label	Yes	No
lower	Yes	No
upper	Yes	No

Child Elements:

*When parent element is **VALUEMAPLABELRENDERER:***

CALLOUTMARKERSYMBOL	Yes	No
CHARTSYMBOL	Yes	No
RASTERSHIELDSYMBOL	Yes	No
SHIELDSYMBOL	Yes	No
TEXTSYMBOL	Yes	No

*When parent element is **VALUEMAPRENDERER:***

GRADIENTFILLSYMBOL	Yes	No
HASHLINESYMBOL	Yes	No
RASTERFILLSYMBOL	Yes	No
RASTERMARKERSYMBOL	Yes	No
SIMPLELINESYMBOL	Yes	No
SIMPLEMARKERSYMBOL	Yes	No
SIMPLEPOLYGONSYMBOL	Yes	No
TRUETYPEMARKERSYMBOL	Yes	No

RASTER_INFO Image - YES ArcMap - YES

No Attributes

Child Elements:

BANDS	Yes	Yes

RASTERFILLSYMBOL Image - YES ArcMap - YES

Attributes:

antialiasing	Yes	No	
image	Yes	Yes	Acetate layer only in ArcMap Server.
overlap	Yes	No	
transparency	Yes	No	
url	Yes	Yes	Acetate layer only in ArcMap Server.

No Child Elements

RASTERMARKERSYMBOL	**Image - YES**	**ArcMap - YES**	
Attributes:			
antialiasing	Yes	No	
hotspot	Yes	No	
image	Yes	Yes	Acetate layer only in ArcMap Server.
overlap	Yes	No	
shadow	Yes	No	
size	Yes	Yes	Acetate layer only in ArcMap Server.
transparency	Yes	No	
url	Yes	Yes	Acetate layer only in ArcMap Server.
usecentroid	Yes	No	

No Child Elements

RASTERSHIELDSYMBOL	**Image - YES**	**ArcMap - NO**
Attributes:		
antialiasing	Yes	No
boundary	Yes	No
font	Yes	No
fontcolor	Yes	No
fontsize	Yes	No
fontstyle	Yes	No
image	Yes	No
labelmode	Yes	No
printmode	Yes	No
shadow	Yes	No
textposition	Yes	No

transparency	Yes	No
url	Yes	No

No Child Elements

REQUEST	**Image - YES**	**ArcMap - YES**

No Attributes

Child Elements:

GET_EXTRACT	Yes	No
GET_FEATURES	Yes	Yes
GET_GEOCODE	Yes	No
GET_IMAGE	Yes	Yes
GET_LAYOUT	No	Yes
GET_RASTER_INFO	Yes	Yes
GET_SERVICE_INFO	Yes	Yes

RESPONSE	**Image - YES**	**ArcMap - YES**

No Attributes

Child Elements:

EXTRACT	Yes	No
FEATURES	Yes	Yes
GEOCODE	Yes	No
IMAGE	Yes	Yes
LAYOUT	No	Yes
RASTER_INFO	Yes	Yes

SERVICEINFO	Yes	Yes

RING	**Image - YES**	**ArcMap - YES**
No Attributes		
Child Elements:		
COORDS	Yes	Yes
HOLE	Yes	Yes
POINT	Yes	Yes

SCALE	**Image - NO**	**ArcMap - YES**
Attributes:		
rf	No	Yes
x	No	Yes
y	No	Yes
No Child Elements		

SCALEBAR	**Image - YES**	**ArcMap - YES**
Attributes:		
antialiasing	Yes	No
barcolor	Yes	Yes
bartransparency	Yes	No
barwidth	Yes	Yes
coords	Yes	Yes
distance	Yes	No
font	Yes	Yes
fontcolor	Yes	Yes
fontsize	Yes	Yes
fontstyle	Yes	Yes
mapunits	Yes	Yes
mode	Yes	No
outline	Yes	No
overlap	Yes	No
precision	Yes	No
round	Yes	No

scaleunits	Yes	Yes
screenlength	Yes	Yes
textransparency	Yes	No

No Child Elements

SCALEDEPENDENTRENDERER	**Image - YES**	**ArcMap - NO**
Attributes:		
lower	Yes	No
upper	Yes	No
Child Elements:		
GROUPRENDERER	Yes	No
SCALEDEPENDENTRENDERER	Yes	No
SIMPLELABELRENDERER	Yes	No
SIMPLERENDERER	Yes	No
VALUEMAPLABELRENDERER	Yes	No
VALUEMAPRENDERER	Yes	No

SCREEN	**Image - YES**	**ArcMap - YES**
Attributes:		
dpi	Yes	Yes

No Child Elements

SDEWORKSPACE	**Image - YES**	**ArcMap - NO**
Attributes:		
database	Yes	No

encrypted	Yes	No
geoindexdir	Yes	No
instance	Yes	No
name	Yes	No
password	Yes	No
server	Yes	No
user	Yes	No

No Child Elements

SEPARATORS	**Image - YES**	**ArcMap - YES**

Attributes:

cs	Yes	Yes
ts	Yes	Yes

No Child Elements

SERVICEINFO	**Image - YES**	**ArcMap - YES**

No Attributes

Child Elements:

*When parent element is **RESPONSE for an ArcMap Image Service and the dataframe attribute is used in GET_SERVICE_INFO:***

DATAFRAMEINFO	No	Yes
LAYOUTINFO	No	Yes

*When parent element is **RESPONSE for an Image or Feature Service:***

ENVIRONMENT	Yes	Yes
LAYERINFO	Yes	Yes
PROPERTIES	Yes	Yes

SHAPEWORKSPACE	**Image - YES**	**ArcMap - NO**

Attributes:

codepage	Yes	No
directory	Yes	No
geoindexdir	Yes	No

name	Yes	No
shared	Yes	No

No Child Elements

SHIELDSYMBOL	**Image - YES**	**ArcMap - NO**
Attributes:		
antialiasing	Yes	No
font	Yes	No
fontcolor	Yes	No
fontsize	Yes	No
fontstyle	Yes	No
labelmode	Yes	No
minsize	Yes	No
shadow	Yes	No
type	Yes	No

No Child Elements

SIMPLELABELRENDERER	**Image - YES**	**ArcMap - NO**
Attributes:		
featureweight	Yes	No
field	Yes	No
howmanylabels	Yes	No
labelbufferratio	Yes	No
labelpriorities	Yes	No
labelweight	Yes	No
linelabelposition	Yes	No
rotationalangles	Yes	No
Child Elements:		
CALLOUTMARKERSYMBOL	Yes	No
CHARTSYMBOL	Yes	No
RASTERSHIELDSYMBOL	Yes	No
SHIELDSYMBOL	Yes	No

TEXTSYMBOL	Yes	No	

SIMPLELINESYMBOL	**Image - YES**	**ArcMap - YES**	
Attributes:			
antialiasing	Yes	No	
captype	Yes	No	
color	Yes	Yes	Acetate layer only in ArcMap Server.
jointype	Yes	No	
overlap	Yes	No	
transparency	Yes	No	
type	Yes	Yes	Acetate layer only in ArcMap Server. Known problems with dot, dash_dot, and dash_dot_dot in ArcMap Server.
width	Yes	Yes	Acetate layer only in ArcMap Server.

No Child Elements

SIMPLEMARKERSYMBOL	**Image - YES**	**ArcMap - YES**	
Attributes:			
antialiasing	Yes	No	
color	Yes	Yes	Acetate layer only in ArcMap Server.
outline	Yes	Yes	Acetate layer only in ArcMap Server.
overlap	Yes	No	
shadow	Yes	No	
transparency	Yes	No	
type	Yes	Yes	Acetate layer only in ArcMap Server. Triangle and star don't show outline correctly in ArcMap Server.
usecentroid	Yes	No	
width	Yes	Yes	Acetate layer only in ArcMap Server.

No Child Elements

SIMPLEPOLYGONSYMBOL	Image - YES	ArcMap - YES	
Attributes:			
antialiasing	Yes	No	Acetate layer only in ArcMap Server.
boundary	Yes	Yes	Acetate layer only in ArcMap Server.
boundarycaptype	Yes	No	
boundarycolor	Yes	Yes	Acetate layer only in ArcMap Server.
boundaryjointype	Yes	No	
boundarytransparency	Yes	No	
boundarytype	Yes	Yes	Acetate layer only in ArcMap Server.
boundarywidth	Yes	Yes	Acetate layer only in ArcMap Server.
fillcolor	Yes	Yes	Acetate layer only in ArcMap Server.
fillinterval	Yes	Yes	Acetate layer only in ArcMap Server.
filltransparency	Yes	No	
filltype	Yes	Yes	Acetate layer only in ArcMap Server.
overlap	Yes	No	
transparency	Yes	No	

No Child Elements

SIMPLERENDERER	Image - YES	ArcMap - NO
No Attributes		
Child Elements:		
GRADIENTFILLSYMBOL	Yes	No
HASHLINESYMBOL	Yes	No
RASTERFILLSYMBOL	Yes	No

RASTERMARKERSYMBOL	Yes	No
SIMPLELINESYMBOL	Yes	No
SIMPLEMARKERSYMBOL	Yes	No
SIMPLEPOLYGONSYMBOL	Yes	No
TRUETYPEMARKERSYMBOL	Yes	No

SPATIALFILTER	**Image - YES**	**ArcMap - YES**
Attributes:		
relation	Yes	Yes
Child Elements:		
BUFFER	Yes	Yes
ENVELOPE	Yes	Yes
MULTIPOINT	Yes	Yes
POLYGON	Yes	Yes
POLYLINE	Yes	Yes

SPATIALQUERY	**Image - YES**	**ArcMap - YES**
Attributes:		
accuracy	Yes	No
featurelimit	Yes	Yes
joinexpression	Yes	No
jointables	Yes	No
searchorder	Yes	Yes
subfields	Yes	Yes
where	Yes	Yes
Child Elements:		

*When parent element is **BUFFER:***			
SPATIALFILTER	Yes	Yes	
*When parent element is **GET_FEATURES:***			
BUFFER	Yes	Yes	
FEATURECOORDSYS	Yes	Yes	
FILTERCOORDSYS	Yes	Yes	
SPATIALFILTER	Yes	Yes	
*When parent element is **LAYER and LAYERDEF in GET_IMAGE:***			
BUFFER	Yes	Yes	
SPATIALFILTER	Yes	Yes	

TARGETLAYER	**Image - YES**	**ArcMap - YES**	
Attributes:			
id	Yes	Yes	
No Child Elements			

TEXT	**Image - YES**	**ArcMap - YES**	
Attributes:			
coords	Yes	Yes	
label	Yes	Yes	
Child Elements:			
TEXTMARKERSYMBOL	Yes	Yes	

TEXTMARKERSYMBOL	**Image - YES**	**ArcMap - YES**	
Attributes:			
angle	Yes	No	
antialiasing	Yes	No	
blockout	Yes	Yes	Acetate layer only in ArcMap Server.

font	Yes	Yes	Acetate layer only in ArcMap Server.
fontcolor	Yes	Yes	Acetate layer only in ArcMap Server.
fontsize	Yes	Yes	Acetate layer only in ArcMap Server.
fontstyle	Yes	Yes	Acetate layer only in ArcMap Server.
glowing	Yes	No	
halignment	Yes	Yes	Acetate layer only in ArcMap Server.
interval	Yes	No	
outline	Yes	Yes	Acetate layer only in ArcMap Server.
overlap	Yes	No	
printmode	Yes	No	
transparency	Yes	No	
valignment	Yes	Yes	Acetate layer only in ArcMap Server.

No Child Elements

TEXTSYMBOL	**Image - YES**	**ArcMap - NO**
Attributes:		
antialiasing	Yes	No
blockout	Yes	No
font	Yes	No
fontcolor	Yes	No
fontsize	Yes	No
fontstyle	Yes	No
glowing	Yes	No
interval	Yes	No
outline	Yes	No
printmode	Yes	No
shadow	Yes	No
transparency	Yes	No

No Child Elements

TOC	**Image - NO**	**ArcMap - YES**

No Attributes

Child Elements:

TOCGROUP	No	Yes

TOCCLASS	**Image - NO**	**ArcMap - YES**
Attributes:		
description	No	Yes
label	No	Yes

No Child Elements

TOCGROUP	**Image - NO**	**ArcMap - YES**
Attributes:		
heading	No	Yes

Child Elements:

TOCCLASS	No	Yes

TRUETYPEMARKERSYMBOL	**Image - YES**	**ArcMap - YES**	
Attributes:			
angle	Yes	Yes	Acetate layer only in ArcMap Server.
anglefield	Yes	No	
antialiasing	Yes	No	
character	Yes	Yes	Acetate layer only in ArcMap Server.
font	Yes	Yes	Acetate layer only in ArcMap Server.
fontcolor	Yes	Yes	Acetate layer only in ArcMap Server.
fontsize	Yes	Yes	Acetate layer only in ArcMap Server.
fontstyle	Yes	Yes	Acetate layer only in ArcMap Server.
glowing	Yes	No	
outline	Yes	Yes	Acetate layer only in ArcMap Server.

overlap	Yes	No
rotatemethod	Yes	No
shadow	Yes	No
transparency	Yes	No
usecentroid	Yes	No

No Child Elements

UIFONT	**Image - YES**	**ArcMap - YES**
Attributes:		
color	Yes	Yes
name	Yes	Yes
size	Yes	Yes
style	Yes	Yes

No Child Elements

VALUEMAPLABELRENDERER	**Image - YES**	**ArcMap - NO**
Attributes:		
featureweight	Yes	No
howmanylabels	Yes	No
labelbufferratio	Yes	No
labelfield	Yes	No
labelpriorities	Yes	No
labelweight	Yes	No
linelabelposition	Yes	No
lookupfield	Yes	No
rotationalangles	Yes	No
Child Elements:		
EXACT	Yes	No
OTHER	Yes	No
RANGE	Yes	No

VALUEMAPRENDERER	**Image - YES**	**ArcMap - NO**
Attributes:		

lookupfield	Yes	No

Child Elements:

EXACT	Yes	No
OTHER	Yes	No
RANGE	Yes	No

WORKSPACES **Image - YES** **ArcMap - NO**

No Attributes

Child Elements:

*When parent element is **MAP in a map configuration file, GET_IMAGE, GET_EXTRACT:***

IMAGEWORKSPACE	Yes	No
SDEWORKSPACE	Yes	No
SHAPEWORKSPACE	Yes	No

Using GET_LAYOUT and LAYOUT with ArcMap Image Services

Introduction

The purpose of GET_LAYOUT and its response LAYOUT is to generate an ArcMap layout and provide the location and filename of the layout.

ArcMap documents contain one or more data frames. The data frames can be viewed as a map or in a layout. Most of the sample requests that follow are based on an ArcMap document consisting of two data frames. The Layers data frame has five layers from the ESRIDATA dataset. The following table summarizes the layer names, shapefile name, and layer ID number.

Layers data frame Layer Name	Shapefile Name	Layer ID
Cities	CITIES	0
Provinces	PROVINCE	1
States	STATES	2
Countries	CNTRY94	3
Ocean	WORLD30	4

The second data frame is used in some examples. This data frame is named States and contains the following layers:

States data frame Layer Name	Shapefile Name	Layer ID
Cities	CITIES	0
Roads	ROADS	1
States	STATES	2

GET_LAYOUT Request and LAYOUT Response

ArcMap layouts can be viewed using GET_LAYOUT. The layout format and size are determined by the ArcMap document and cannot be changed using GET_LAYOUT. However, GET_LAYOUT can request data in a data frame at different envelopes and projections. This request is for display only and can be used to generate a layout in one of several formats. The simplest GET_LAYOUT request requires no child elements. With this request, a layout is generated exactly as established in the ArcMap Image Service.

Simple GET_LAYOUT request:

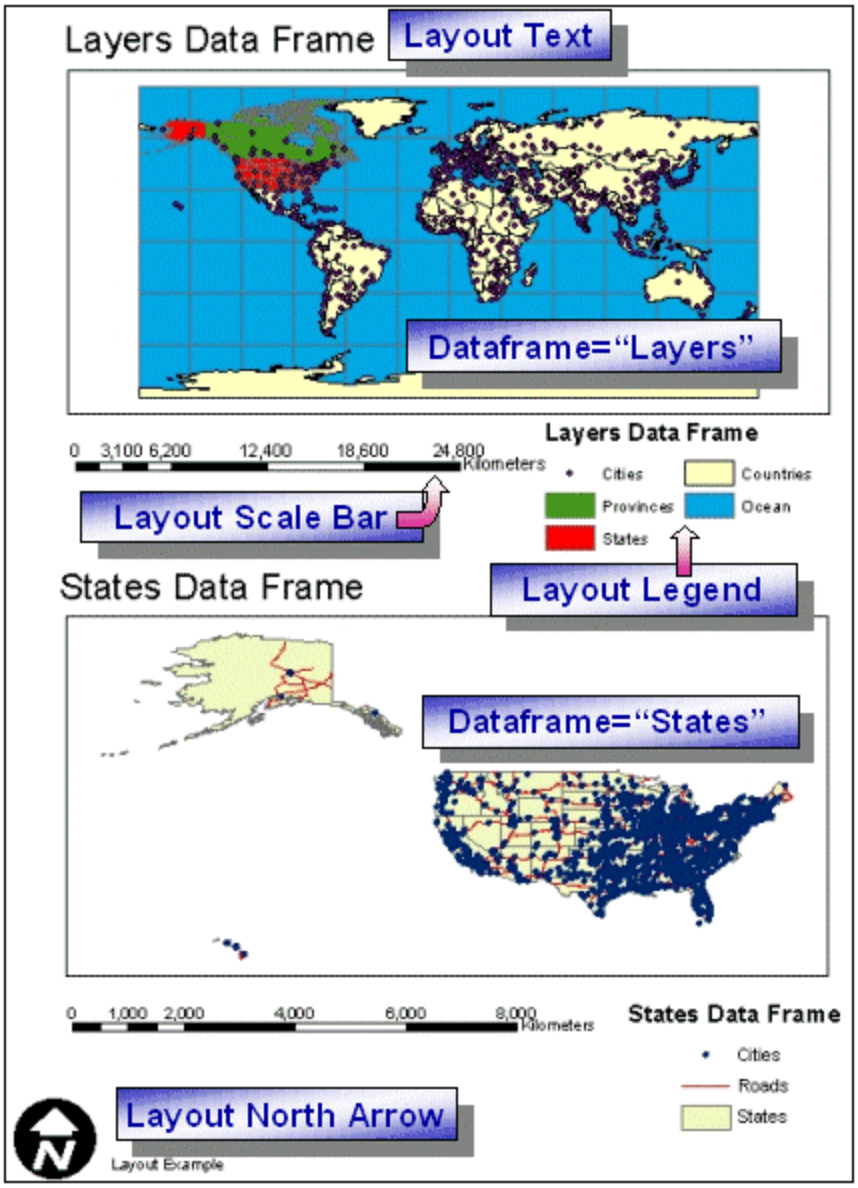
```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_LAYOUT>
    </GET_LAYOUT>
  </REQUEST>
</ARCXML>
```

The LAYOUT response is similar to an IMAGE response and includes a default envelope and the name and location of the layout. One significant difference is that the ENVELOPE coordinates are in the page units of a layout and not in map units.

LAYOUT response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <LAYOUT>
      <ENVELOPE minx="-3.08333333333333" miny="0"
maxx="11.5833333333333" maxy="11" />
      <OUTPUT
url="http://mycomputer.domain.com/output/world_MYCOMPUTER2102209.png"
file="c:\arcims\output\world_MYCOMPUTER2102209.png" />
    </LAYOUT>
  </RESPONSE>
</ARCXML>
```

The layout is returned using the formatting defined in the ArcMap document. This format cannot be altered. The layout always contains at least the active data frame. It can also contain multiple data frames and other margin information such as text, north arrows, scale bars, and legends. None of these features can be generated in the GET_LAYOUT request.



GET_LAYOUT has two child elements: PROPERTIES and DATAFRAME. PROPERTIES defines general characteristics of the layout as a whole, such as the extent of the layout to show, a default projection for all data frames, and the output format of the layout. DATAFRAME provides information on an individual data frame, such as the extent to show and what projection to use. The following example shows the framework of a GET_LAYOUT request when these child elements are included.

GET_LAYOUT request with PROPERTIES and DATAFRAME:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_LAYOUT>
      <PROPERTIES>
      </PROPERTIES>
```

```
      <DATAFRAME id="Layers" >
      </DATAFRAME>
      <DATAFRAME id="States" >
      </DATAFRAME>
    </GET_LAYOUT>
  </REQUEST>
</ARCXML>
```

Using ENVELOPE

ENVELOPE can be used in both PROPERTIES and DATAFRAME.

When ENVELOPE is used in PROPERTIES, the envelope coordinates are referencing the layout as a whole in page units. When a user pans or zooms on a layout, what really happens is that the extent of the layout changes, not the extent of any data frames. When ENVELOPE is used in DATAFRAME, the extent of the data in the data frame changes. These units are in map units.

In the following example, ENVELOPE is included in both PROPERTIES and in both DATAFRAMEs. In the PROPERTIES section, the extent is unchanged from the service. However, if a user were to pan or zoom the layout, these envelope coordinates would be the ones to change. In the DATAFRAME section, the map in the Layers data frame has a new extent covering North America. The map in the States data frame has a new extent showing the eastern United States.

Using ENVELOPE in PROPERTIES and DATAFRAME:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_LAYOUT>
      <PROPERTIES>
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
      </PROPERTIES>
      <DATAFRAME id="Layers" >
        <ENVELOPE minx="-144" miny="19" maxx="-52" maxy="81" />
      </DATAFRAME>
      <DATAFRAME id="States" >
        <ENVELOPE minx="-94" miny="32" maxx="-73" maxy="46" />
      </DATAFRAME>
    </GET_LAYOUT>
  </REQUEST>
</ARCXML>
```

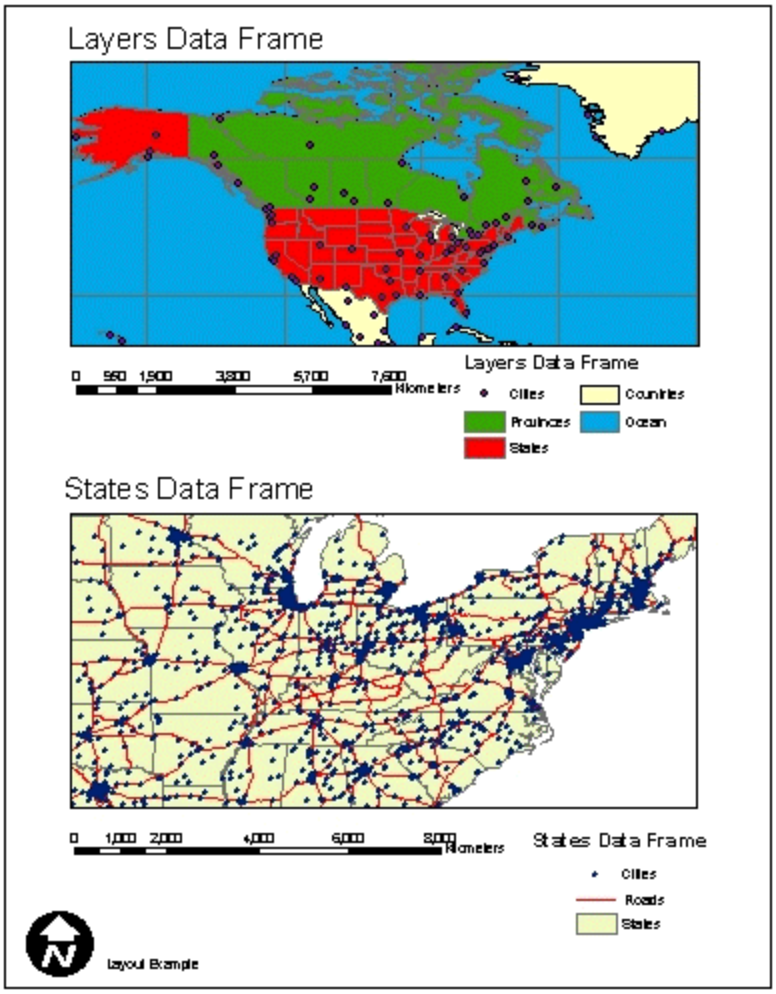
The ENVELOPE in the response is in layout page units.

LAYOUT response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <LAYOUT>
```

```
      <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11"/>
      <OUTPUT
url="http://mycomputer.domain.com/output/world_MYCOMPUTER2102209.png"
file="c:\arcims\output\world_MYCOMPUTER2102209.png" />
    </LAYOUT>
  </RESPONSE>
</ARCXML>
```

The layout is generated with new extents for the Layers and States data frames.



Using SCALE

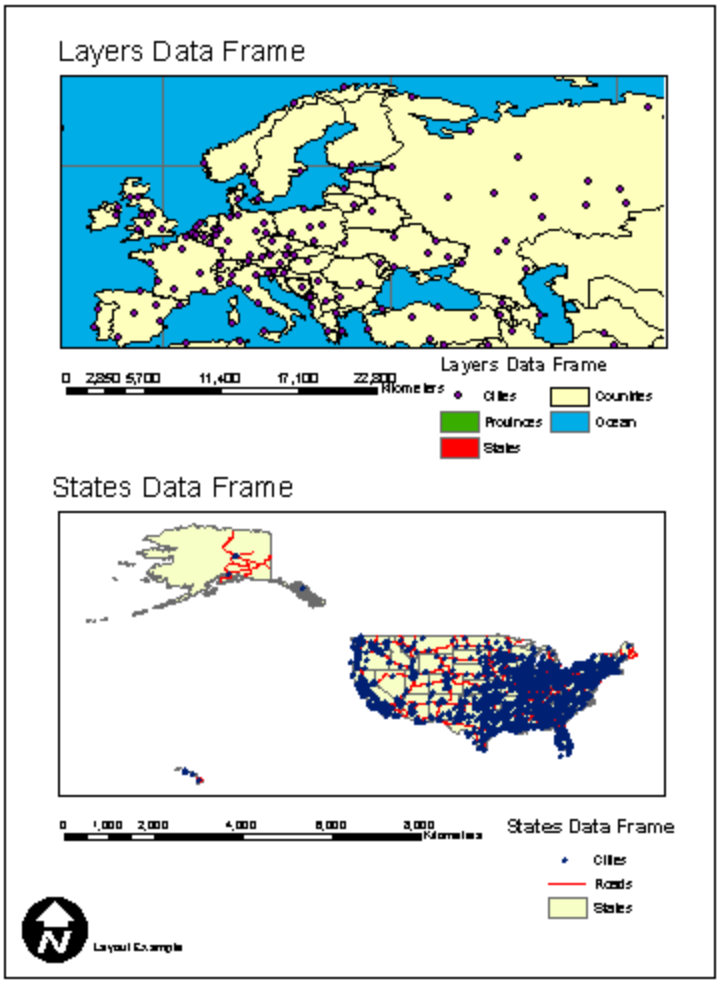
Inside DATAFRAME, SCALE can be used instead of ENVELOPE. SCALE defines a relative scale and the center point of the map. In the next example, the scale factor for the Layers data frame is set to 1:15000000 and is centered on Berlin, Germany. Note that in the attribute *rf*, the value is "15000000" and not "1:15000000".

Using SCALE in DATAFRAME:

```
<?xml version="1.0" encoding="UTF8"?>
```

```
<ARCXML version="1.1">
<REQUEST>
  <GET_LAYOUT>
    <PROPERTIES>
        <ENVELOPE minx="0" miny="0" maxx="11" maxy="8.5" />
      </PROPERTIES>
      <DATAFRAME id="Layers">
        <SCALE rf="15000000" x="13.3" y="52.8" />
      </DATAFRAME>
    </GET_LAYOUT>
  </REQUEST>
</ARCXML>
```

The layout is generated with a new scale for the "Layers" data frame.



Changing the Layout Format and Output Size using OUTPUT, LAYOUT, GET_LAYOUT autoresize

A layout can be requested in one of the following formats: AI, BMP, EMF, EPS, JPG, PDF, PNG8, or PNG24. The format is specified using OUTPUT inside PROPERTIES. If OUTPUT is not included, the default output format is JPG.

BMP, JPG, PNG8, and PNG24 formats

During a request, IMAGESIZE in PROPERTIES sets the size of layouts in BMP, JPG, PNG8, and PNG24 formats. If IMAGESIZE is not used, the default image size is 400 x 300 pixels.

The output size is controlled using IMAGESIZE *width* and *height*. The maximum size of an image can be no greater than the image memory limit set when an ArcMap Image Service is started. For example, an image memory limit of 1 MB allows a layout image no larger than 262,144 pixels (512 x 512) to be generated. By default, a layout image is limited in size to 4 MB, which corresponds to 1,048,576 pixels. Changes can be made to the image size and pixel count using the ArcIMS Administrator. For more information, see *Using ArcIMS*.

A requested layout greater than the maximum pixel count can be reduced in size to within the maximum pixel count by using GET_LAYOUT *autoresize*. If *autoresize* is set to "true", a requested layout is reduced in size to always be within the maximum pixel count. If *autoresize* is not included, and the requested image is too big, an ERROR message is returned.

In the next example, a layout is requested in PNG8 format: OUTPUT *type="PNG8"*. The requested layout size is 1600 x 1200 pixels: IMAGESIZE *width="1600" height="1200"*. In order to trap for image requests that are too large, GET_LAYOUT *autoresize* is set to "true".

GET_LAYOUT request with OUTPUT, IMAGESIZE, and autoresize="true":

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_LAYOUT autoresize="true">
      <PROPERTIES>
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
        <IMAGESIZE width="1600" height="1200" />
        <OUTPUT type="PNG8" />
      </PROPERTIES>
      <DATAFRAME id="Layers" >
        <ENVELOPE minx="-144" miny="19" maxx="-52" maxy="81" />
      </DATAFRAME>
      <DATAFRAME id="States" >
        <ENVELOPE minx="-94" miny="32" maxx="-73" maxy="46" />
      </DATAFRAME>
    </GET_LAYOUT>
  </REQUEST>
</ARCXML>
```

The LAYOUT response includes a resized image, since the requested image size was greater than allowed. In OUTPUT, the attributes *height* and *width* show the new image size. The generated layout is in PNG format.

IMAGE response with resized image information:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <LAYOUT>
      <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
      <OUTPUT
url="http://mycomputer.domain.com/output/world_MYCOMPUTER3633699.png
width="1182" height="886"
file="c:\arcims\output\world_MYCOMPUTER3633699.png" />
    </LAYOUT>
  </RESPONSE>
</ARCXML>
```

AI, EMF, EPS, and PDF formats

AI, EMF, EPS, and PDF formats do not have the same image size restrictions. When these formats are requested, the output size matches the actual layout size in page units. Therefore, a layout in an ArcMap document set as "C" size is generated as "C" size during the GET_LAYOUT request. However, if IMAGESIZE is used in a request, the height and width are still used to calculate the image memory limit. To get around this problem, remove IMAGESIZE from the request or set GET_LAYOUT *autoresize* to "true".

Using OUTPUT to Control Layout Names and Locations

OUTPUT, in addition to setting the output format, provides a way to define the location and filename of the output layout. By default, the output directory and URL location are determined at the time the service is started. When a request is made, the ArcIMS Spatial Server assigns a filename. The name includes the service name, the computer name the layout was generated on, and a randomly generated number. If the service is named "world" and the computer is "MYCOMPUTER", then an example filename is world_MYCOMPUTER1248849.png.

When output is included in the request, it overrides information from when the service was started. The output files are not automatically deleted by ArcIMS Tasker Windows service or UNIX daemon. In order for the files to be deleted, the *taskfile* property must be set in tasker.properties. This property points to a file named Tasks.xml by default. If you plan to use OUTPUT *type* most of the time in GET_LAYOUT requests, you should use the Tasks.xml file. Both tasker.properties and Tasks.xml are found in the

AppServer/Tasker directory located within the ArcIMS install directory.
The following steps are recommended to ensure that GET_LAYOUT file output is being deleted.

- In addition to OUTPUT *type* in a GET_LAYOUT request, also include a directory location and/or file name for output files. This directory should be different than the default Output directory.

  <OUTPUT path="c:\arcims\layout_output" baseurl="http://mycomputer.domain.com/arcims/layout_output" />

  Information about assigning path and filenames is covered in the next two sections.
- In Tasks.xml, add the following line:

  <CLEANUP dir="c:/ArcIMS/layout_output" exts="ai, bmp, emf, eps, jpg, pdf, png" interval="10" />

  where
  dir is the directory location of the layout output directory
  exts are the file extensions to be deleted
  interval is the amount of time in minutes to delete files.

For information on using Tasker property files, see the ArcIMS Help.

OUTPUT works with paired attributes for defining a file output location and/or filename. If one of the attributes is used, its pair is required. Determining which pair to use depends on whether the ArcIMS Spatial Server defines the output filename or you do. In all cases, you specify the output directory and URL.

Attribute	Paired Attribute	Filename assignment	Example: http://mycomputer/arcims/...
path	baseurl	ArcIMS assigns random filename.	world_MYCOMPUTER1248849.png
name	url	You assign a filename.	myfilename.png

When starting a service or including OUTPUT in the request, UNC pathnames are valid. For example, instead of using "c:\arcims\layout_output", "\\myComputer\arcims\layout_output" can be used.

ArcIMS Spatial Server assigns filename

In the following example, the ArcIMS Spatial Server assigns a filename, but the user assigns the directory and URL. The attribute pair in this scenario is *path-baseurl*.

OUTPUT when ArcIMS Spatial Server defines filename:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_LAYOUT>
      <PROPERTIES>
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
        <OUTPUT type="pdf" path="c:\arcims\layout_output"
baseurl="http://mycomputer.domain.com/layout_output" />
      </PROPERTIES>
    </GET_LAYOUT>
  </REQUEST>
</ARCXML>
```

LAYOUT response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <LAYOUT>
      <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
      <OUTPUT
url="http://mycomputer.domain.com/layout_output/world_MYCOMPUTER2983738.pdf"
file="c:\arcims\layout_output\world_MYCOMPUTER2983738.pdf" />
    </LAYOUT>
  </RESPONSE>
</ARCXML>
```

User assigns filename

In the next example, the user assigns a filename, directory, and URL. The attribute pair in this scenario is *name-url*.

OUTPUT when user defines filename:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_LAYOUT>
      <PROPERTIES>
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
        <OUTPUT type="pdf"
name="c:\arcims\layout_output\layoutimage.pdf"
url="http://mycomputer.domain.com/layout_output/layoutimage.pdf" />
      </PROPERTIES>
    </GET_LAYOUT>
  </REQUEST>
</ARCXML>
```

LAYOUT response:

```
<?xml version="1.0" encoding="UTF8"?>
```

```
<ARCXML version="1.1">
  <RESPONSE>
    <LAYOUT>
      <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
      <OUTPUT
url="http://mycomputer.domain.com/layout_output/layoutimage.pdf"
file="c:\arcims\layout_output\layoutimage.pdf" />
    </LAYOUT>
  </RESPONSE>
</ARCXML>
```

Restricting OUTPUT

When using an ArcIMS HTML Viewer, ArcIMS Java Viewer, ArcExplorer 4, or any other client using the ArcIMS Servlet Connector, the OUTPUT element is restricted by default.

- In a GET_LAYOUT request, OUTPUT is ignored.

GET_LAYOUT request when OUTPUT is restricted:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_LAYOUT>
      <PROPERTIES>
        <OUTPUT name="c:\arcims\layout_output\layoutimage.png"
url="http://mycomputer.domain.com/layout_output/layoutimage.png"
/>
      </PROPERTIES>
    </GET_LAYOUT>
  </REQUEST>
</ARCXML>
```

Note that in this situation, the only format available for a layout is the format declared when the service was started, either *.JPG or *.PNG.

- In an LAYOUT response, the attribute *file* is not returned. The OUTPUT location is the default location specified when the ArcIMS Image Service was started. The ArcIMS Spatial Server assigns the filename. OUTPUT includes only *url*.

IMAGE response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <LAYOUT>
      <ENVELOPE minx="-180" miny="-144" maxx="180" maxy="144" />
      <OUTPUT
url="http://mycomputer.domain.com/layout_output/world_MYCOMPUTER2102209.png"
/>
    </LAYOUT>
```

```
  </RESPONSE>
</ARCXML>
```

These restrictions can be lifted by setting the properties *spatialServer.AllowRequestOutput* and *spatialServer.AllowResponsePath* to true in Esrimap_prop. This property file is found in the servlet engine directory. For more information on the location of Esrimap_prop and its properties, see the ArcIMS Help.

These restrictions apply only when the ArcIMS Servlet Connector is used. They do not apply to the ActiveX, ColdFusion, or Java Connectors. The OUTPUT filename extensions are limited to *.ai, *.bmp, *.emf, *.eps, *.jpg, *.pdf, and *.png, regardless of whether OUTPUT is restricted or not.

Using Projections with GET_LAYOUT

The projection of data in data frames can be changed using the projection elements:

- FEATURECOORDSYS
- FILTERCOORDSYS

For a complete discussion on the different elements, refer to Using Projection Elements.

FILTERCOORDSYS is used to specify the coordinate system of the requesting client or the coordinates in ENVELOPE in a DATAFRAME. FEATURECOORDSYS is used to specify to which coordinate system the data in the data frames should be transformed. All the examples so far for have been in geographic coordinates (decimal degrees), which have an ID of "4326".

FILTERCOORDSYS and FEATURECOORDSYS can be used inside PROPERTIES and DATAFRAME. When used inside PROPERTIES, all data frames are transformed to the same projection. In the next example, all data frames in the layout are requested in Robinson, which has an ID of "54030". FILTERCOORDSYS and FEATURECOORDSYS are included only in the PROPERTIES section.

Using FILTERCOORDSYS and FEATURECOORDSYS inside PROPERTIES:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_LAYOUT>
      <PROPERTIES>
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
        <FEATURECOORDSYS id="54030" />
        <FILTERCOORDSYS id="54030" />
      </PROPERTIES>
      <DATAFRAME id="Layers" >
      </DATAFRAME>
      <DATAFRAME id="States" >
      </DATAFRAME>
```

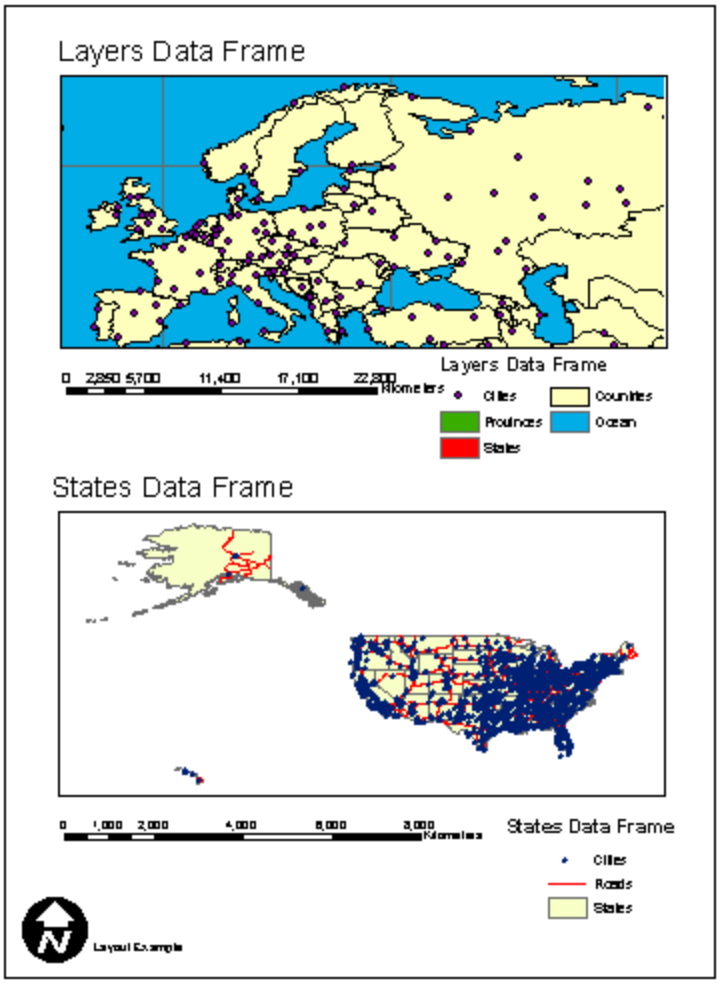
```
    </GET_LAYOUT>
  </REQUEST>
</ARCXML>
```

In the response, the ENVELOPE coordinates are still in layout page units.

IMAGE response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <LAYOUT>
      <ENVELOPE minx="0" miny="-0.45" maxx="8.5" maxy="11.45" />
      <OUTPUT
url="http://mycomputer.domain.com/output/world_MYCOMPUTER2102209.jpg"
file="c:\arcims\output\world_MYCOMPUTER2102209.jpg" />
    </LAYOUT>
  </RESPONSE>
</ARCXML>
```

The returned map image shows both the Layers and States data frames in the Robinson coordinate system.



Using the Robinson coordinate system with the "States" data frame does not produce the best results. In order to set the projection for individual data frames, FILTERCOORDSYS and FEATURECOORDSYS can be used inside DATAFRAME in addition to PROPERTIES. When used in a DATAFRAME, projection information in the PROPERTIES section is ignored for that data frame.

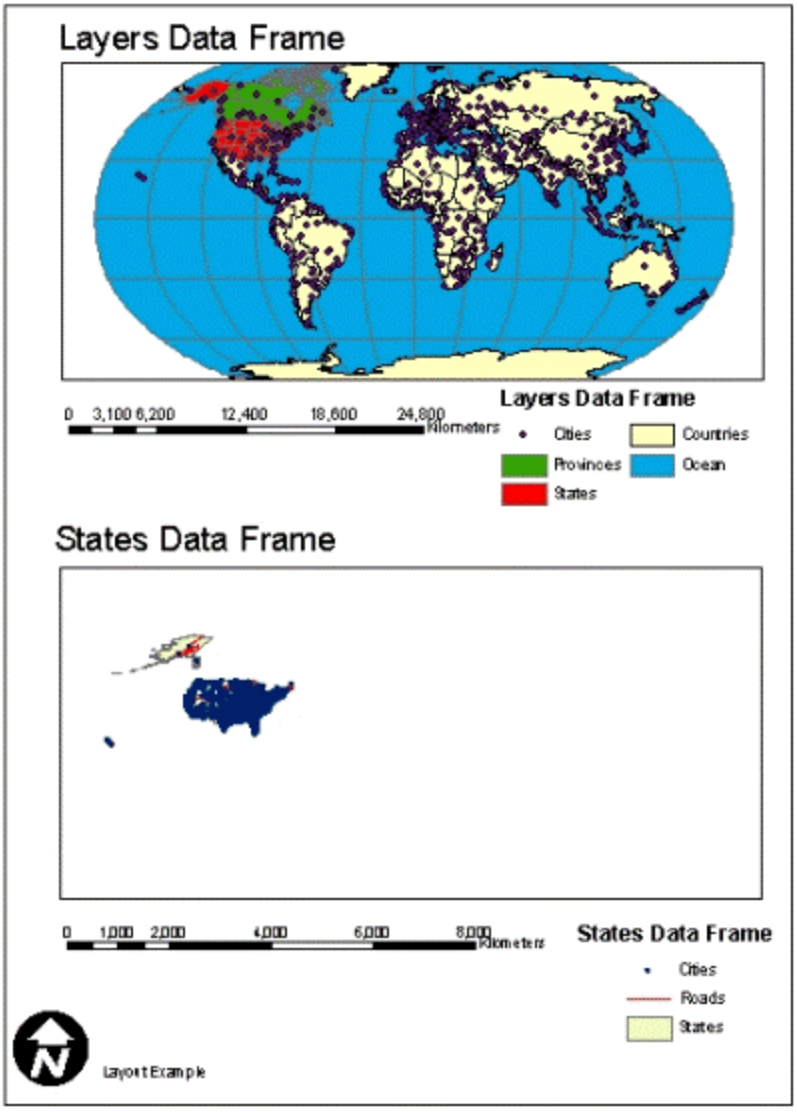
In the next example, FILTERCOORDSYS and FEATURECOORDSYS are used in the "States" data frame and are set to North America Albers Equal Area Conic (102008). To summarize this request, FILTERCOORDSYS and FEATURECOORDSYS in the PROPERTIES section are set to Robinson (54030). All data frames use this projection information unless FILTERCOORDSYS and FEATURECOORDSYS are also used in DATAFRAME. The "Layers" data frame has no projection information, and the data is transformed to Robinson. The "States" data frame does have projection information, and the data is transformed to North America Albers rather than to Robinson.

ENVELOPE should also be included in DATAFRAME when using projection elements. The ENVELOPE coordinates should be in the same projection as FILTERCOORDSYS. If ENVELOPE is not included, the coordinates are assumed to be in the projection of the service.

Using FILTERCOORDSYS and FEATURECOORDSYS inside PROPERTIES and DATAFRAME:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_LAYOUT>
      <PROPERTIES>
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
        <FEATURECOORDSYS id="54030" />
        <FILTERCOORDSYS id="54030" />
      </PROPERTIES>
      <DATAFRAME id="Layers" >
      </DATAFRAME>
      <DATAFRAME id="States" >
        <ENVELOPE minx="-8260377.77928749" miny="-3192427.8093698"
maxx="3697151.36816421" maxy="5775719.05121898" />
        <FEATURECOORDSYS id="102008" />
        <FILTERCOORDSYS id="102008" />
      </DATAFRAME>
    </GET_LAYOUT>
  </REQUEST>
</ARCXML>
```

The returned image shows the "Layers" data frame in Robinson and the "States" data frame in North America Albers.


Layers Data Frame
0 3,100 6,200 12,400 18,600 24,800
Kilometers
Layers Data Frame
Cities
Countries
Provinces
Ocean
States
States Data Frame
0 1,000 2,000 4,000 6,000 8,000
Kilometers
States Data Frame
Cities
Roads
States
N
Layout Example

Using GET_EXTRACT and EXTRACT

Introduction

The purpose of GET_EXTRACT and EXTRACT is to extract specified layers of an Image Service into a set of shapefiles, yielding one shapefile for each layer. Data can be extracted from shapefiles and ArcSDE vector layers. To use the extract elements, you should understand map configuration file syntax, the relationship between map configuration files and ArcIMS services, how to send a request to the ArcIMS Spatial Server, and how to handle a response.

GET_EXTRACT requests are usually used with the Image Server. Although data can be extracted from a Feature Service, GET_EXTRACT must be made the same way it is for an Image Service. In the Java Viewers, the request must be made independently of the applets and the Java Viewer API. A request to the Extract Server cannot be made from ArcExplorer 4. GET_EXTRACT requests are not valid with the Feature or ArcMap Servers.

The request syntax for GET_EXTRACT is very much like GET_IMAGE. Many of the combinations of elements supported by GET_IMAGE are also supported by GET_EXTRACT. For a more detailed review of GET_IMAGE, see Using GET_IMAGE and IMAGE.

A GET_EXTRACT request identifies which layers to extract, the extract envelope, and any query constraints to be included in the request. GET_EXTRACT does not support BUFFER or LEGEND. Once the request is processed, the data is compressed into a ZIP file, and the response contains the location of the ZIP file.

The typical scenario for using the Extract Server involves setting up an HTML Viewer with an Image Service that contains an extract extension. GET_IMAGE requests are sent to the Image Service to manipulate the map and query data. When it is time to extract data, a GET_EXTRACT request is sent to the same Image Service but the request is routed to the Extract Server. The details of this scenario are covered below.

Creating a Map Configuration File with an Extract Extension

In order to trigger the Extract Server, an extract extension must be included with at least one layer in the map configuration file. The following map configuration file contains six layers: World, Countries, States, Provinces, World Cities, and US Cities. Each layer includes rendering information. In addition, the WORLD30, CNTRY94, and US Cities layers include an extract extension. Details about the extension are given below.

Figure 1: An example map configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>

<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP dynamic="true">
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="C:\ESRIDATA\WORLD"
/>
        <SHAPEWORKSPACE name="shp_ws-2" directory="C:\ESRIDATA\USA" />
        <SHAPEWORKSPACE name="shp_ws-3" directory="C:\ESRIDATA\CANADA"
/>
      </WORKSPACES>
      <LAYER type="featureclass" name="WORLD30" visible="true"
id="Ocean">
        <DATASET name="WORLD30" type="polygon" workspace="shp_ws-0" />
        <EXTENSION type="extract" >
          <EXTRACTPARAMS clip="true" />
        </EXTENSION>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="0,153,255"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="CNTRY94" visible="true"
id="Countries">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <EXTENSION type="extract" >
          <EXTRACTPARAMS clip="true" />
        </EXTENSION>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid"
fillcolor="255,255,153"/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="STATES" visible="true"
id="States">
        <DATASET name="STATES" type="polygon" workspace="shp_ws-2" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="province" visible="true"
id="Provinces">
        <DATASET name="province" type="polygon" workspace="shp_ws-3" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="0,153,0" />
        </SIMPLERENDERER>
```

```
      </LAYER>
      <LAYER type="featureclass" name="World Cities" visible="true"
id="Cities">
        <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
      <SPATIALQUERY where="POPULATION > 300000" subfields="NAME
POPULATION"  />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="102,0,102" width="8.0" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="US Cities" visible="true"
id="35">
        <DATASET name="CITIES" type="point" workspace="shp_ws-2" />
        <SPATIALQUERY where="POP1990 &gt; 100000" subfields="CITY_NAME
STATE_NAME POP1990 MALES FEMALES"  />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="0,255,0" type="star" width="8.0"
/>
        </SIMPLERENDERER>
        <EXTENSION type="extract" >
        <EXTRACTPARAMS clip="true" >
          <OUTPUTFILE file="us_cities" >
            <OUTPUTFIELD name="CITY_NAME" alias="City" />
            <OUTPUTFIELD name="STATE_NAME" alias="State" />
            <OUTPUTFIELD name="POP1990" alias="Population" />
            <OUTPUTFIELD name="MALES" alias="Male_pop" />
            <OUTPUTFIELD name="FEMALES" alias="Female_pop" />
          </OUTPUTFILE>
        </EXTRACTPARAMS>
      </EXTENSION>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

The examples in the remainder of this document use this map configuration file started as an Image Service.

Using the extract EXTENSION

In the map configuration file, the US Cities layer contains an extract EXTENSION. This extension notifies the ArcIMS Spatial Server that the Extract Server will be used. It also provides a place to customize the shapefile name and specify which attributes should be included.

```
<EXTENSION type="extract" >
  <EXTRACTPARAMS clip="true" >
    <OUTPUTFILE file="us_cities" >
      <OUTPUTFIELD name="CITY_NAME" alias="City" />
      <OUTPUTFIELD name="STATE_NAME" alias="State" />
      <OUTPUTFIELD name="POP1990" alias="Population" />
      <OUTPUTFIELD name="MALES" alias="Male_pop" />
      <OUTPUTFIELD name="FEMALES" alias="Female_pop" />
    </OUTPUTFILE>
  </EXTRACTPARAMS>
```

```
</EXTENSION>
```

The different extract elements are:

- EXTRACTPARAMS (required) is the main element and must always be included when EXTENSION is used. An option is available to clip to the layers at the current extent boundary in the viewer using the *clip* attribute. When *clip* is set to true in the map configuration file, features are clipped at the extent of the viewer. However, none of a layer's database values are prorated for any clipped features. If clipping is not used, features partially within the envelope are extracted in their entirety.

  The default for *clip* is "false". If you do not want a layer to be clipped, the EXTENSION and EXTRACTPARAMS elements are not required for the layer. In the map configuration file, STATES, Province, and World Cities do not include these elements. The layers will be extracted, but features will not be clipped.

  In order to clip the features in a layer, the EXTENSION and EXTRACTPARAMS elements must be included, and *clip* must be set to "true". In the map configuration file, the WORLD30, CNTRY94, and World Cities layers include the extension elements, and *clip* is set to "true".

- OUTPUTFILE (optional) is used to name the extracted shapefile. File names must use all English characters. In the above example, the shapefile is named us_cities.

- OUTPUTFIELD (optional) can be used to set an alias name for a field in the database. Shapefiles use DBF files to store attribute data, and a field in the DBF is limited to 10 characters. Since other databases often allow more than 10 characters, it's possible to end up with two fields in the DBF file with the same name. ArcIMS does not allow shapefiles with duplicate field names to be used.

  OUTPUTFIELD allows the addition of an alias name to replace the actual field name. In this example, when the data is extracted for the US Cities layer, the DBF file contains "City" instead of "CITY_NAME" for the first field, and so on.

  To limit the extracted fields to those listed in OUTPUTFIELD, the attribute subfields must be included in a SPATIALQUERY (or QUERY). In the map configuration file, *subfields* is used in the following line:

```
<SPATIALQUERY where="POP1990 &gt; 100000" subfields="CITY_NAME
STATE_NAME POP1990 MALES FEMALES"  />
```

As noted above, EXTRACTPARAMS is required, while OUTPUTFILE and OUTPUTFIELD are optional. In order to use the Extract Server, at least one layer (any layer) must include the extract extension. The minimum information needed is the following:

```
      <EXTENSION type="extract" >
```

```
        <EXTRACTPARAMS clip="true" />
      </EXTENSION>
```

Naming extracted shapefiles

Two different methods can be used to name extracted shapefiles. The naming scheme is set up in the map configuration file.

1. The default method for naming extracted shapefiles is to use the name of each layer's *id* attribute value. ArcIMS Author always assigns an ID using numbers. If you want to name your shapefile something more meaningful, you can assign a character string to the *id*. For example, for the WORLD30 layer, the extracted shapefile is named "Ocean".
2. The second method is to name a shapefile using OUTPUTFILE in a layer EXTENSION.

Using SPATIALQUERY in a map configuration file

SPATIALQUERY can be used to set both attribute and spatial constraints on a layer in a map configuration file. In the above map configuration file, the World Cities layer has a constraint to show only cities with a population greater than 300,000, and the US Cities layer has a constraint to show cities with a population greater than 100,000. (You could also use QUERY as QUERY can be used to set constraints on attributes. However, you are encouraged to use SPATIALQUERY since it can handle both attribute and spatial queries.) Only data that meets the query criteria is extracted. SPATIALFILTER can also be used with SPATIALQUERY to limit data extraction to that within the specified filter.

Routing to the Extract Server

The Extract Server is a private server. GET_EXTRACT requests are made to an Image or Feature Service and must be routed to the Extract Server. This routing information is contained in the URL that is sent to the ArcIMS site such as in the following example (all one line):

http://myComputer.domain.com/servlet/com.esri.esrimap.Esrimap?ClientVersion=3.1
&ServiceName=myservice
&CustomService=Extract
&Form=True&Encode=True

Using the GET_EXTRACT Request

Once the map configuration file has been started as an Image Service, GET_EXTRACT requests can be made. The following example is a typical request to the Extract Server.

A typical GET_EXTRACT request:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_EXTRACT>
      <PROPERTIES>
        <ENVELOPE minx="-130" miny="30" maxx="-90" maxy="60" />
      </PROPERTIES>
    </GET_EXTRACT>
  </REQUEST>
</ARCXML>
```

In general, although ENVELOPE is not required, it should always be included in the request to indicate the extent of the data to extract. If ENVELOPE is not included, all data is extracted. In the above request, the GET_EXTRACT request takes features from the six layers within the defined envelope and extracts them with their attributes to six shapefiles. The shapefiles are named Ocean, Countries, States, Provinces, Cities, and us_cities. Although rendering is included in the map configuration file, no rendering information is retained when the shapefiles are generated.

The EXTRACT Response

The EXTRACT response includes the envelope and the location of the ZIP file.

An EXTRACT response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <EXTRACT>
      <ENVELOPE minx="-130.000000" miny="30.000000" maxx="-90.000000"
maxy="60.000000" />
      <OUTPUT file="c:\arcims\output\myservice_mymachine33934311.zip"
url="http://mymachine.domain.com/output/myservice_mymachine33934311.zip"
/>
    </EXTRACT>
  </RESPONSE>
</ARCXML>
```

In the above example, the shapefiles are zipped into a file named myservice_mymachine33934311.zip. This file is located on the server in the directory c:\arcims\output. The file can be accessed from a Web site using the URL http://mymachine.domain.com/output/myservice_mymachine33934311.zip.

Customizing the GET_EXTRACT Request

Constraints can be added to a GET_EXTRACT request in order to override information in the service. The following examples show different methods for extracting subsets of data using:

- IMAGESIZE
- LAYERLIST and LAYERDEF
- SPATIALQUERY (or QUERY)
- LAYER
- OUTPUT

IMAGESIZE

In a GET_EXTRACT request, IMAGESIZE is used to calculate which layers should be extracted based on any scale dependencies. If a layer is out of range based on the scale, it is not extracted. If IMAGESIZE is not used in a request, the default image size used for calculating the extent is 400 x 300 pixels.

When IMAGESIZE is used in GET_IMAGE requests, the same IMAGESIZE should be used in any GET_EXTRACT requests. If you find that a different set of layers is extracted compared to the list of layers in a map image, double-check that IMAGESIZE is the same for both GET_IMAGE and GET_EXTRACT.

In GET_EXTRACT, the only valid IMAGESIZE attributes are *height* and *width*. In the following example, *height* and *width* are set to 600 and 800, respectively.

Using IMAGESIZE in a GET_EXTRACT request:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_EXTRACT>
      <PROPERTIES>
        <ENVELOPE minx="-130" miny="30" maxx="-90" maxy="60" />
        <IMAGESIZE height="600" width="800" />
      </PROPERTIES>
    </GET_EXTRACT>
  </REQUEST>
</ARCXML>
```

LAYERLIST and LAYERDEF

Layers in GET_EXTRACT can be switched on and off using the LAYERDEF element. Layers that are set to *visible="false"* are not extracted. In the following example, data for the States and Provinces layers is not extracted since the attribute *visible* has been set to "false".

Using LAYERDEF in a GET_EXTRACT request:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_EXTRACT>
    <PROPERTIES>
      <ENVELOPE minx="-130" miny="30" maxx="-90" maxy="60" />
      <LAYERLIST>
        <LAYERDEF id="Ocean" visible="true" />
```

```
        <LAYERDEF id="Countries" visible="true" />
        <LAYERDEF id="States" visible="false" />
        <LAYERDEF id="Provinces" visible="false" />
        <LAYERDEF id="Cities" visible="true" />
        <LAYERDEF id="35" visible="true" />
      </LAYERLIST>
    </PROPERTIES>
  </GET_EXTRACT>
</REQUEST>
</ARCXML>
```

The attribute *nodefault* can be used with LAYERLIST in a GET_EXTRACT request. When *nodefault* is set to "true", only the layers listed in the LAYERLIST are extracted. Only layers of type featureclass can be extracted; any acetate and image layers in the LAYERLIST are ignored. The following example will produce the same results as the previous example. The changes are that *nodefault* is set to true in LAYERLIST, and the LAYERDEF information for States and Provinces has been removed.

Using LAYERLIST nodefault in a GET_EXTRACT request:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_EXTRACT>
    <PROPERTIES>
      <ENVELOPE minx="-130" miny="30" maxx="-90" maxy="60" />
      <LAYERLIST nodefault="true">
        <LAYERDEF id="Ocean" visible="true" />
        <LAYERDEF id="Countries" visible="true" />
        <LAYERDEF id="Cities" visible="true" />
        <LAYERDEF id="35" visible="true" />
      </LAYERLIST>
    </PROPERTIES>
  </GET_EXTRACT>
</REQUEST>
</ARCXML>
```

SPATIALQUERY

SPATIALQUERY is used to set a spatial constraint on a layer. In the following example, a SPATIALQUERY for the Cities layer sets an envelope to a smaller extent than the ENVELOPE in PROPERTIES. The response includes only cities with a population greater than 300,000 within the newly defined envelope. Queries in a request cannot override a constraint already set in a service. The service limits cities to those with a population greater than 300,000.

Using SPATIALQUERY with an envelope in a GET_EXTRACT request:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_EXTRACT>
    <PROPERTIES>
      <ENVELOPE minx="-130" miny="30" maxx="-90" maxy="60" />
```

```
      <LAYERLIST>
        <LAYERDEF id="Ocean" visible="true" />
        <LAYERDEF id="Countries" visible="true" />
        <LAYERDEF id="States" visible="false" />
        <LAYERDEF id="Provinces" visible="false" />
        <LAYERDEF id="Cities" visible="true" >
          <SPATIALQUERY>
            <SPATIALFILTER relation="area_intersection">
              <ENVELOPE minx="-120" miny="40" maxx="-100" maxy="60" />
            </SPATIALFILTER>
          </SPATIALQUERY>
        </LAYERDEF>
        <LAYERDEF id="35" visible="true" />
      </LAYERLIST>
    </PROPERTIES>
  </GET_EXTRACT>
</REQUEST>
</ARCXML>
```

SPATIALQUERY can also be used in a GET_EXTRACT request:

- To set additional constraints on a layer based on attribute data in a specified field. (QUERY can also be used but SPATIALQUERY is recommended.) In the next example, SPATIALQUERY in the Countries layer will extract only countries beginning with the letter "C".
- To limit which fields are extracted using the *subfields* attribute. In the States layer, only the fields STATE_NAME and POP1990 are extracted.

Using SPATIALQUERY on attribute data in a GET_EXTRACT request:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_EXTRACT>
    <PROPERTIES>
      <ENVELOPE minx="-130" miny="30" maxx="-90" maxy="60" />
      <LAYERLIST>
        <LAYERDEF id="Ocean" visible="true" />
        <LAYERDEF id="Countries" visible="true" >
          <SPATIALQUERY where="NAME LIKE 'C%'" />
        </LAYERDEF>
        <LAYERDEF id="States" visible="true" >
          <SPATIALQUERY subfields="STATE_NAME POP1990" />
        </LAYERDEF>
        <LAYERDEF id="Provinces" visible="false" />
        <LAYERDEF id="Cities" visible="true" />
        <LAYERDEF id="35" visible="true" />
      </LAYERLIST>
    </PROPERTIES>
  </GET_EXTRACT>
</REQUEST>
</ARCXML>
```

Dynamic layers using LAYER

Dynamic layers not in the map configuration file can be included in the list of layers to

extract using LAYER. The dynamic layer must include a DATASET in order to be extracted. Acetate and image layers cannot be extracted.

To add a new layer, a WORKSPACES section must be included in the map configuration file or the request. It is recommended to always include WORKSPACES in the map configuration file since references to path names on the host computer are included. By keeping all WORKSPACES in the map configuration file, the directory locations remain hidden from users.

In the next example, Rivers is located in the shp_ws-0 workspace, the same workspace as Ocean, Countries, and Cities. The example also includes the dynamic layer made up of a selected set from an existing layer in the service. Both of these dynamic layers can be extracted. The Ocean, Countries, World Cities, Rivers, and selected layers are extracted.

Including a dynamic LAYER for extraction:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_EXTRACT>
      <PROPERTIES>
        <ENVELOPE minx="-147" miny="-46" maxx="-43" maxy="70" />
        <LAYERLIST>
          <LAYERDEF id="Ocean" visible="true" />
          <LAYERDEF id="Countries" visible="true" />
          <LAYERDEF id="States" visible="false" />
          <LAYERDEF id="Provinces" visible="false" />
          <LAYERDEF id="Cities" visible="true" />
          <LAYERDEF id="35" visible="false" />
        </LAYERLIST>
      </PROPERTIES>
      <LAYER type="featureclass" name="Rivers" visible="true"
id="Rivers">
        <DATASET name="RIVERS" type="line" workspace="shp_ws-0" />
      </LAYER>
      <LAYER type="featureclass" name="Selected Countries"
visible="true" id="Selected">
        <DATASET fromlayer="countries" />
        <SPATIALQUERY where="NAME=&amp;apos;Brazil&amp;apos;" />
      </LAYER>
    </GET_EXTRACT>
  </REQUEST>
</ARCXML>
```

Dynamic layers and LAYERLIST

The attribute *nodefault* can be used with LAYERLIST to extract only specified layers in the service and specified dynamic layers. In the next example, Ocean, Countries, World Cities, and Rivers are extracted. If a layer is not in the LAYERLIST, it is not extracted.

Specifying layers for extraction:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<ARCXML version="1.1">
  <REQUEST>
    <GET_EXTRACT>
      <PROPERTIES>
        <ENVELOPE minx="-147" miny="-46" maxx="-43" maxy="70" />
        <LAYERLIST nodefault="true">
          <LAYERDEF id="Ocean" visible="true" />
          <LAYERDEF id="Countries" visible="true" />
          <LAYERDEF id="Cities" visible="true" />
          <LAYERDEF id="Rivers" visible="true" />
        </LAYERLIST>
      </PROPERTIES>
      <LAYER type="featureclass" name="Rivers" visible="true"
id="Rivers">
        <DATASET name="RIVERS" type="line" workspace="shp_ws-0" />
      </LAYER>
      <LAYER type="featureclass" name="Selected Countries"
visible="true" id="Selected">
        <DATASET fromlayer="countries" />
        <SPATIALQUERY where="NAME=&amp;apos;Brazil&amp;apos;" />
      </LAYER>
    </GET_EXTRACT>
  </REQUEST>
</ARCXML>
```

OUTPUT

OUTPUT, when used in a GET_EXTRACT request, defines the location and filename of the ZIP file. By default, the output directory and URL location are determined at the time the Image Service is started. When a request is made, the ArcIMS Spatial Server assigns a filename. The name includes the service name, the computer name the image was generated on, and a randomly generated number. If the service is named "world" and the computer is "MYCOMPUTER", then an example ZIP file is "world_MYCOMPUTER1248849.zip". The ArcIMS Tasker Windows service or UNIX daemon automatically deletes the ZIP files on a user-specified interval.

OUTPUT can be used in both a map configuration file and in a request. When used in a map configuration file, the OUTPUT information overrides the output information stored when the service is started. When used in a request, OUTPUT overrides information in both the map configuration file and when the service is started. When OUTPUT is used, the output files are not automatically deleted by ArcIMS Tasker. In order for the files to be deleted, the *taskfile* property must be set in tasker.properties. For information on setting this property, see the topic "Setting up ArcIMS Monitor and ArcIMS Tasker Property Files" in the System Administration: Administering Property Files chapter of the ArcIMS Help Contents.

OUTPUT works with paired attributes. If one of the attributes is used, its pair is also required. The attribute pairs are listed in the table below.

Attribute	Paired Attribute	Filename Assignment

path	baseurl	ArcIMS assigns random filename.
name	url	User assigns a filename.

If *path* and *baseurl* are used, output files can be redirected to a new directory but ArcIMS assigns the filename. In the example below, the new output directory is c:\arcims\newdirectory, and the new URL is http://mymachine.domain.com/newdirectory.

Using OUTPUT path and baseurl in a GET_EXTRACT request:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_EXTRACT>
      <PROPERTIES>
        <ENVELOPE minx="-130" miny="30" maxx="-90" maxy="60" />
        <OUTPUT path="c:\arcims\newdirectory"
baseurl="http://mycomputer.domain.com/newdirectory" />
      </PROPERTIES>
    </GET_EXTRACT>
  </REQUEST>
</ARCXML>
```

Based on the above request, a ZIP file is generated and placed in c:\arcims\newdirectory. The ArcIMS Spatial Server generated the filename, which is myservice_mymachine2286774.zip.

EXTRACT response when OUTPUT path and baseurl are used:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <EXTRACT>
      <ENVELOPE minx="-130.000000" miny="30.000000" maxx="-90.000000"
maxy="60.000000" />
      <OUTPUT file="c:\arcims\newdirectory\myservice_mymachine2286774.zip"
url="http://mymachine.domain.com/newdirectory/myservice_mymachine2286774.zip"
/>
    </EXTRACT>
  </RESPONSE>
</ARCXML>
```

If the attribute pair *name* and *url* is used, an output filename must be included along with the path. In the next example, the file is called "myzipfile.zip".

Using OUTPUT name and url in a GET_EXTRACT request:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_EXTRACT>
      <PROPERTIES>
```

```
        <ENVELOPE minx="-130" miny="30" maxx="-90" maxy="60" />
        <OUTPUT name="c:\arcims\newdirecotory\myzipfile.zip"
url="http://mycomputer.domain.com/newdirectory/myzipfile.zip" />
      </PROPERTIES>
    </GET_EXTRACT>
  </REQUEST>
</ARCXML>
```

In this response, a ZIP file named myzipfile.zip is generated and placed in c:\arcims\newdirectory. The URL is http://mymachine.domain.com/newdirectory/myzipfile.zip.

EXTRACT response when OUTPUT name and url are used:

```
Figure 3: An EXTRACT Response. <?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <EXTRACT>
      <ENVELOPE minx="-130.000000" miny="30.000000" maxx="-90.000000"
maxy="60.000000" />
      <OUTPUT file="c:\arcims\newdirectory\myzipfile.zip"
url="http://mymachine.domain.com/newdirectory/myzipfile.zip" />
    </EXTRACT>
  </RESPONSE>
</ARCXML>
```

Restricting OUTPUT

When using an ArcIMS HTML Viewer, ArcIMS Java Viewer, ArcExplorer 4, or any other browser using the ArcIMS Servlet Connector, the OUTPUT element is restricted by default.

- In a GET_EXTRACT request, OUTPUT is ignored.

  GET_EXTRACT request when OUTPUT is restricted:

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  <ARCXML version="1.1">
    <REQUEST>
      <GET_EXTRACT>
        <PROPERTIES>
          <ENVELOPE minx="-130" miny="30" maxx="-90" maxy="60" />
          <OUTPUT name="c:\arcims\newdirecotory\myzipfile.zip"
  url="http://mycomputer.domain.com/newdirectory/myzipfile.zip" />
        </PROPERTIES>
      </GET_EXTRACT>
    </REQUEST>
  </ARCXML>
  ```

- In an EXTRACT response, the attribute *file* is not returned. The OUTPUT location is the default location specified when the service was started. The ArcIMS Spatial Server assigns the filename. OUTPUT includes only *url* and not *file*.

EXTRACT response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <EXTRACT>
      <ENVELOPE minx="-130.000000" miny="30.000000" maxx="-90.000000"
maxy="60.000000" />
      <OUTPUT
url="http://mymachine.domain.com/output/myservice_mymachine33934311.zip"
/>
    </EXTRACT>
  </RESPONSE>
</ARCXML>
```

These restrictions can be lifted by setting the properties *spatialServer.AllowRequestOutput* and *spatialServer.AllowResponsePath* to "true" in esrimap_prop. This property file is found in the servlet engine directory. For more information on the location of esrimap_prop and its properties, see the topic "Administering the ArcIMS Servlet Connector" in the System Administration chapter of the Using ArcIMS Help Contents.

These restrictions apply only when the ArcIMS Servlet Connector is used. They do not apply to the ActiveX, ColdFusion, or Java Connectors. The OUTPUT filename extension is restricted to *.zip, regardless of whether OUTPUT is restricted or not.

Transforming Geometry with the Extract Server

The coordinate system of extracted shapefiles can be changed on the fly using the following elements:

- COORDSYS
- FEATURECOORDSYS
- FILTERCOORDSYS

For a complete discussion on the different elements, refer to Using Projection Elements.

COORDSYS is used when adding dynamic LAYERS and no *.prj file is associated with a shapefile or coverage or no spatial references table is available in ArcSDE. FILTERCOORDSYS is used to specify the coordinate system of the requesting client. All the examples so far have been in geographic coordinates (decimal degrees) with an *id="4326"*. FEATURECOORDSYS is used to specify the coordinate system to which the service should be transformed.

In the next example, FILTERCOORDSYS is in geographic coordinates with an *id="4326"*. Note that the envelope coordinates used for the extraction are in geographic coordinates. The extracted shapefiles are requested in Robinson, so FEATURECOORDSYS is set to *id="54030"*.

Using FILTERCOORDSYS and FEATURECOORDSYS in a GET_EXTRACT request:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_EXTRACT>
      <PROPERTIES>
        <ENVELOPE minx="-130" miny="30" maxx="-90" maxy="60" />
        <FILTERCOORDSYS id="4326" />
        <FEATURECOORDSYS id="54030" />
      </PROPERTIES>
    </GET_EXTRACT>
  </REQUEST>
</ARCXML>
```

In the EXTRACT response, the location of the ZIP file is returned along with the ENVELOPE in Robinson coordinates ("54030").

EXTRACT response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <EXTRACT>
      <ENVELOPE minx="-11777530.8282994" miny="3205046.64322241" maxx="-
6782838.54770184" maxy="6329105.54868597" />
      <OUTPUT
url="http://mymachine.domain.com/output/myservice_mymachine33934311.zip"
/>
    </EXTRACT>
  </RESPONSE>
</ARCXML>
```

Using GET_FEATURES and FEATURES

Introduction

GET_FEATURES is a versatile request that can be used to access geometric information about selected items, the attribute results of a query, or both. The information is returned in a FEATURES response.

GET_FEATURES requests can be made on Image, ArcMap Image, and Feature Services. The following map configuration file is used for the GET_FEATURES examples in this document. The file consists of five layers from the ESRIDATA data set. The following table summarizes the layer names, shapefile names, and layer ID numbers.

Layer Name	Shapefile Name	Layer ID
Ocean	WORLD30	0
Countries	CNTRY94	1
States	STATES	2
Provinces	PROVINCE	3
Cities	CITIES	4

Map configuration file used with the GET_FEATURES requests that follow:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP dynamic="true" >
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="C:\ESRIDATA\WORLD"
/>
        <SHAPEWORKSPACE name="shp_ws-2" directory="C:\ESRIDATA\USA" />
        <SHAPEWORKSPACE name="shp_ws-3" directory="C:\ESRIDATA\CANADA"
/>
      </WORKSPACES>
      <LAYER type="featureclass" name="Ocean" visible="true" id="0">
        <DATASET name="WORLD30" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="0,153,255"
```

```
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Countries" visible="true"
id="1">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid"
fillcolor="255,255,153"/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="States" visible="true" id="2">
        <DATASET name="STATES" type="polygon" workspace="shp_ws-2" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Provinces" visible="true"
id="3">
        <DATASET name="province" type="polygon" workspace="shp_ws-3" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="0,153,0" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Cities" visible="true" id="4">
        <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="102,0,102" width="8.0" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

GET_FEATURES can query only one layer of an ArcIMS service at a time. Queries cannot be made on dynamic layers. The request must include either the SPATIALQUERY or QUERY element. The query can extract data based on attribute data, a spatial filter, or a combination of the two. SPATIALQUERY is recommended for all queries since it can handle both attribute and spatial queries.

The following example is a typical GET_FEATURES request. The LAYER has been identified by its ID, which must match the layer ID in the map configuration file. In this example, *id="4"* refers to the Cities layer. The SPATIALQUERY includes both an attribute query and a spatial filter. The attribute query is for cities with more than 10 million people. The spatial query limits the extent to an area that covers North America.

GET_FEATURES request with LAYER and SPATIALQUERY:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES featurelimit="25" beginrecord="0" outputmode="xml"
geometry="false" envelope="true" globalenvelope="true">
      <LAYER id="4" />
      <SPATIALQUERY subfields="#ALL#" where="POPULATION > 10000000" >
```

```
        <SPATIALFILTER relation="area_intersection">
          <ENVELOPE minx="-129" miny="16" maxx="-50" maxy="62"/>
        </SPATIALFILTER>
      </SPATIALQUERY>
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

The information returned in the FEATURES response varies depending on the attributes selected in GET_FEATURES. Each of the attributes is discussed in more detail later in this document.

The following example is the response to the above request. Two cities, Mexico City and New York City, meet the criteria set forth in the SPATIALQUERY. The response includes the location of each city using ENVELOPE. The requested subfields, NAME and POPULATION, are also included. The response also contains the total number of features returned in FEATURECOUNT and a global extent that encompasses both cities in the last ENVELOPE listed.

FEATURES response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <FEATURES>
      <FEATURE>
        <ENVELOPE minx="-99.127571105957" miny="19.4270458221436"
maxx="-99.127571105957" maxy="19.4270458221436"/>
        <FIELDS NAME="Mexico City" POPULATION="14100000"
#SHAPE#="[Geometry]" />
      </FEATURE>
      <FEATURE>
        <ENVELOPE minx="-74.0999984741211" miny="40.75" maxx="-
74.0999984741211" maxy="40.75"/>
        <FIELDS NAME="New York" POPULATION="16472000"
#SHAPE#="[Geometry]" />
      </FEATURE>
      <FEATURECOUNT count="2" hasmore="false" />
      <ENVELOPE minx="-99.127571105957" miny="19.4270458221436" maxx="-
74.0999984741211" maxy="40.75"/>
    </FEATURES>
  </RESPONSE>
</ARCXML>
```

Accessing an ArcIMS Service using GET_FEATURES

Based on attribute settings in a GET_FEATURES request, the FEATURES response returns data in a feature stream or in ArcXML format. Requests to Feature Services are sent to the Feature Server and use the feature streaming format. The stream is a compressed binary format that can be interpreted only by the ArcIMS Java Viewers or ArcMap. The binary stream that is returned includes both feature and attribute data. GET_FEATURES requests to Image and ArcIMS Image Services are limited to data in

ArcXML format. When querying an Image or ArcMap Image Service, two requests must be made to get both a map and attribute data. When using Image Services, requests for a map go to the Image Server using GET_IMAGE. Requests for attribute data go to the Query Server using GET_FEATURES. In order to send a request to the Query Server, the request must be "rerouted". The routing is done in the URL when the request is sent. An example URL is (all one line):

http://mycomputer.domain.com/servlet/com.esri.esrimap.Esrimap
?ClientVersion=3.1
&ServiceName=MyService
&CustomService=Query
&Form=True&Encode=True

When using ArcMap Image Services, both GET_IMAGE and GET_FEATURES requests are sent to the ArcMap Server. No rerouting is needed for GET_FEATURES requests.

The table below summarizes the differences when using GET_FEATURES on Feature and Image Services:

	Feature Services	**Image Services**	**ArcMap Image Services**
GET_FEATURES request uses:	Feature Streaming	ArcXML	ArcXML
FEATURES response uses:	Feature Streaming	ArcXML	ArcXML
GET_FEATURES request for attribute data sent to:	Feature Server	Query Server	ArcMap Server
Request to draw map sent to:	Feature Server using GET_FEATURES	Image Server using GET_IMAGE	ArcMap Server using GET_IMAGE

GET_FEATURES has a series of attributes that are used to format the FEATURES response. The formatting attributes are *outputmode*, *compact*, *geometry*, *skipfeatures*, *envelope*, *attributes*, *checkesc*, and *globalenvelope*. Other attributes are used to keep track of the number of features in the request and response: *beginrecord*, *featurelimit*, *count*, and *hasmore*.

In the examples and explanations below, all requests and responses are in ArcXML format. The requests are made to the map configuration file in the Introduction section, which is assumed to be an Image Service.

Outputmode

The *outputmode* attribute defines whether the data is streamed to the client or sent to the client in ArcXML format.

Outputmode can have one of three values:

- binary
- xml
- newxml

If *outputmode* is binary, then the request sent to the Feature Server and the response are in a compressed binary stream. This is the default and is valid only for Feature Services.

If *outputmode="xml"*, then all features are returned in a short ArcXML format inside the FIELDS element of the FEATURES response.

In the next example, the request is made on the States layer (*id="2"*). The SPATIALQUERY asks for the *subfields* STATE_NAME and SUB_REGION for the State of Washington.

GET_FEATURES request when outputmode="xml":

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES outputmode="xml">
      <LAYER id="2" />
      <SPATIALQUERY subfields="state_name sub_region"
where="STATE_NAME='Washington'" />
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

The response returns the requested information on Washington. The attributes are returned on one line inside the FIELDS element. The attribute names are the same as the field names in the database.

FEATURES response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <FEATURES>
      <FEATURE>
        <FIELDS STATE_NAME="Washington" SUB_REGION="Pacific" />
      </FEATURE>
      <FEATURECOUNT count="1" hasmore="false" />
    </FEATURES>
  </RESPONSE>
</ARCXML>
```

The "xml" *outputmode* can be used only when no XML parser is used since the attribute names in FIELD are always changing. When using an XML parser, *outputmode* shoud be set to "newxml". If *outputmode="newxml"*, then all fields are returned in a longer ArcXML format using the FIELD element. The next request asks for the same information on Washington as the previous request.

GET_FEATURES request when outputmode="newxml":

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES outputmode="newxml">
      <LAYER id="2" />
      <SPATIALQUERY subfields="state_name sub_region"
where="STATE_NAME='Washington'" />
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

The response returns the same information on Washington but each attribute is included inside a separate FIELD element. With this format, the attribute names are standardized in FIELD and do not change.

FEATURES response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <FEATURES>
      <FEATURE>
        <FIELDS>
          <FIELD name="STATE_NAME" value="Washington" />
          <FIELD name="SUB_REGION" value="Pacific" />
        </FIELDS>
      </FEATURE>
      <FEATURECOUNT count="1" hasmore="false" />
    </FEATURES>
  </RESPONSE>
</ARCXML>
```

Geometry and Compact

The attribute *geometry* is used to determine whether feature geometry is returned. The attribute *compact* is a toggle that can be used to return the geometry in a long or short format in the FEATURES response. By default, the geometry is returned in a longer format using POINT. Note that geometry is returned only if "#SHAPE#" or "#ALL#" is included in *subfields* in the SPATIALQUERY.

In the next request, *geometry* is set to "true", and *compact* is set to "false". The "#SHAPE#" field has been added to the list of *subfields* in SPATIALQUERY.

GET_FEATURES request when outputmode="xml", geometry="true", and compact="false":

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES outputmode="xml" geometry="true" compact="false">
      <LAYER id="2" />
      <SPATIALQUERY subfields="#SHAPE# state_name sub_region"
where="STATE_NAME='Washington'" />
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

In the response, the geometry for Washington is returned in a longer format. Each point in the polygon is recorded inside the POINT element.

FEATURES response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
<RESPONSE>
  <FEATURES>
    <FEATURE>
      <FIELDS STATE_NAME="Washington" SUB_REGION="Pacific"
#SHAPE#="[Geometry]" />
      <POLYGON>
        <RING>
          <POINT x="-122.400749" y="48.2253952" />
          <POINT x="-122.461585" y="48.2285423" />
          <POINT x="-122.453155" y="48.1286735" />
          .  .  .  .  .
          <POINT x="-122.400749" y="48.2253952" />
        </RING>
      </POLYGON>
    </FEATURE>
    <FEATURECOUNT count="1" hasmore="false" />
  </FEATURES>
</RESPONSE>
</ARCXML>
```

When *compact="true"*, the response uses COORDS to return the geometry data in a compact format. In the next request, *geometry* and *compact* are set to "true".

GET_FEATURES request when outputmode="xml", geometry="true", and compact="true":

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES outputmode="xml" geometry="true" compact="true">
      <LAYER id="2" />
      <SPATIALQUERY subfields="#SHAPE# state_name sub_region"
where="STATE_NAME='Washington'" />
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

In the response, the coordinates are all grouped together inside the COORDS element. Note that this response uses the default separators between the coordinates and coordinate pairs, which are a space and semicolon, respectively. These separators can be changed by using SEPARATORS in the map configuration file or request.

FEATURES response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <FEATURES>
      <FEATURE>
        <FIELDS STATE_NAME="Washington" SUB_REGION="Pacific"
#SHAPE#="[Geometry]" />
        <POLYGON>
          <RING>
            <COORDS>-122.400749 48.2253952;-122.461585
48.2285423;-122.453155 48.1286735; . . . . .;-122.400749 48.2253952;-
122.461585 48.2285423</COORDS>
          </RING>
        </POLYGON>
      </FEATURE>
      <FEATURECOUNT count="1" hasmore="false" />
    </FEATURES>
  </RESPONSE>
</ARCXML>
```

Beginrecord, Featurelimit, Count, Hasmore

The attributes *beginrecord* and *featurelimit* are used together in GET_FEATURES to limit the number of features returned in the FEATURES response. If too many features are returned, the processing time in the client may become unacceptable.

Featurelimit is used to set the maximum number of features to return. *Beginrecord* is used to start the retrieval at a specified record. For example, if a maximum of seven features should be returned, *featurelimit* is set to "7". For the first retrieval, *beginrecord* starts at "1" (the default). The following example uses *featurelimit* and *beginrecord*.

GET_FEATURES request when featurelimit="7" and beginrecord="1":

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES featurelimit="7" beginrecord="1" outputmode="xml"
geometry="false">
      <LAYER id="2" />
      <SPATIALQUERY subfields="#SHAPE# state_name sub_region"
where="SUB_REGION='Mtn'" />
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

The attributes *count* and *hasmore* in FEATURECOUNT in the FEATURES response contain information on the number of features returned and whether more features are available to extract. To determine whether more features are available to extract, *hasmore* is set to "true" for more records or "false" for no more records. In this example, the number of features returned in *count* is "7", and *hasmore* is set to "true".

FEATURES response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <FEATURES>
      <FEATURE>
        . . . First 6 Records Returned . . .
      </FEATURE>
      <FEATURE>
        <FIELDS STATE_NAME="Arizona" SUB_REGION="Mtn"
#SHAPE#="[Geometry]" />
      </FEATURE>
      <FEATURECOUNT count="7" hasmore="true" />
    </FEATURES>
  </RESPONSE>
</ARCXML>
```

If more features are still available, another GET_FEATURES request should be sent, this time with *beginrecord* set to "8". The developer is responsible for handling additional requests programmatically.

GET_FEATURES request when featurelimit="7" and beginrecord="8":

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES featurelimit="7" beginrecord="8" outputmode="xml"
geometry="false">
      <LAYER id="2" />
      <SPATIALQUERY subfields="#SHAPE# state_name sub_region"
where="SUB_REGION='Mtn'" />
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

In the response, *count* is set to "1", and *hasmore* is set to "false", indicating that no more records need to be extracted.

FEATURES response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <FEATURES>
      <FEATURE>
        <FIELDS STATE_NAME="New Mexico" SUB_REGION="Mtn"
#SHAPE#="[Geometry]" />
```

```
      </FEATURE>
      <FEATURECOUNT count="1" hasmore="false" />
    </FEATURES>
  </RESPONSE>
</ARCXML>
```

Skipfeatures

When the attribute *skipfeatures* is used, no features are returned; only the feature count is returned. *Skipfeatures* is valid only if *outputmode* is "xml" or "newxml". In the following request, a query is made on the Cities layer (*id="4"*) for cities with a population greater than 10 million.

GET_FEATURES request using skipfeatures:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES skipfeatures="true" outputmode="newxml">
      <LAYER id="4" />
      <SPATIALQUERY subfields="#ALL#" where="POPULATION &gt; 10000000"
/>
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

The response includes only the FEATURECOUNT. In this example, 11 cities have a population greater than 10 million.

FEATURES response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <FEATURES>
      <FEATURECOUNT count="11" hasmore="false" />
    </FEATURES>
  </RESPONSE>
</ARCXML>
```

The number of features returned depends on *beginrecord* and *featurelimit*. If *featurelimit* has a value, the feature count will never be larger than that value. In the next example, *featurelimit* is set to "5" on the same request.

GET_FEATURES request using skipfeatures and featurelimit:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES skipfeatures="true" outputmode="newxml"
featurelimit="5" beginrecord="1">
      <LAYER id="4" />
```

```
      <SPATIALQUERY subfields="#ALL#" where="POPULATION &gt; 10000000"
/>
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

In the response, FEATURECOUNT is "5", but *hasmore* is set to "true", indicating that there are more features to extract.

FEATURES response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <FEATURES>
      <FEATURECOUNT count="5" hasmore="true" />
    </FEATURES>
  </RESPONSE>
</ARCXML>
```

To get the total number of features in a layer, a SPATIALQUERY can be specified with an empty *where* clause. Note that even when *skipfeatures* is set to "true", the query is executed and features are actually extracted before they are counted.

GET_FEATURES request using skipfeatures and empty where clause:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES skipfeatures="true" outputmode="newxml" >
      <LAYER id="4" />
      <SPATIALQUERY subfields="#ALL#" where="" />
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

In the response, the total number of features for the layer is included.

FEATURES response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <FEATURES>
      <FEATURECOUNT count="606" hasmore="false" />
    </FEATURES>
  </RESPONSE>
</ARCXML>
```

Envelope, Attributes, and Checkesc

The attribute *envelope* is used to request the bounding envelope of each returned feature.

The attribute *attributes* is used to determine whether the attribute data should be returned. If only geometry is needed, then *attributes* can be set to "false".

The attribute *checkesc* is used if the returned data should include escaped characters for ampersand, single quote, double quote, less than, and greater than. For example, if the value for SUB_REGION in the States layer were "P&NW", then:

- If *checkesc="false"*, in the response the value of SUB_REGION is "P&NW".
- If *checkesc="true"*, the value of SUB_REGION is "P&NW".

In the next request, *attributes*, *envelope*, and *checkesc* are set to "true". The assumption is also made that the SUB_REGION is "P&NW".

GET_FEATURES request when attributes="true", envelope="true", and checkesc="true":

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES outputmode="xml" geometry="false" attributes="true"
envelope="true" checkesc="true">
      <LAYER id="2" />
      <SPATIALQUERY subfields="#SHAPE# state_name sub_region"
where="STATE_NAME='Washington'" />
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

In the response, ENVELOPE contains the bounding extent of Washington. FIELDS contains the attribute information. The value for SUB_REGION has been escaped to "P&NW".

FEATURES response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <FEATURES>
      <FEATURE>
        <ENVELOPE minx="-124.731422" miny="45.5432510" maxx="-
116.918151" maxy="49.0000038"/>
        <FIELDS STATE_NAME="Washington" SUB_REGION="P&amp;NW"
#SHAPE#="[Geometry]" />
      </FEATURE>
      <FEATURECOUNT count="1" hasmore="false" />
    </FEATURES>
  </RESPONSE>
</ARCXML>
```

Globalenvelope

The attribute *globalenvelope* is used to request the bounding envelope of all features returned in the response. Since the number of features returned depends on *beginrecord* and *featurelimit*, only the actual records extracted based on *featurelimit* are included in the overall envelope. Note that in order for the global envelope to be returned, *subfields* in the SPATIALQUERY must include either "#SHAPE#" or "#ALL#".

In the next example, *attributes* is set to "false" and *envelope* and *globalenvelope* are set to "true". A query is made on the States layer for states in the Pacific region.

GET_FEATURES request when attributes="false", envelope="true", and gloabalenvelope="true":

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES outputmode="xml" geometry="false" attributes="false"
envelope="true" globalenvelope="true">
      <LAYER id="2" />
      <SPATIALQUERY subfields="#SHAPE#" where="SUB_REGION='Pacific'" />
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

In the response, an ENVELOPE is included for each FEATURE but no attributes are included for the FEATURE. At the end of the response, the overall bounding ENVELOPE is included.

FEATURES response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <FEATURES>
      <FEATURE>
        <ENVELOPE minx="-124.731422424316" miny="45.5432510375977"
maxx="-116.918151855469" maxy="49.0000038146973"/>
      </FEATURE>
      <FEATURE>
        <ENVELOPE minx="-124.558395385742" miny="41.9877891540527"
maxx="-116.469444274902" maxy="46.2362594604492"/>
      </FEATURE>
      <FEATURE>
        <ENVELOPE minx="-124.391471862793" miny="32.5357246398926"
maxx="-114.124450683594" maxy="42.0023460388184"/>
      </FEATURE>
      <FEATURE>
        <ENVELOPE minx="-160.245178222656" miny="18.9247817993164"
maxx="-154.793869018555" maxy="22.2324924468994"/>
      </FEATURE>
      <FEATURE>
        <ENVELOPE minx="-178.215026855469" miny="51.5844345092773"
```

```
maxx="-129.990539550781" maxy="71.4066467285156"/>
      </FEATURE>
      <FEATURECOUNT count="5" hasmore="false" />
      <ENVELOPE minx="-178.215026855469" miny="18.9247817993164"
maxx="-114.124450683594" maxy="71.4066467285156"/>
    </FEATURES>
  </RESPONSE>
</ARCXML>
```

Transforming Geometry

Both FILTERCOORDSYS and FEATURECOORDSYS can be used with GET_FEATURES to transform data. Data that can be transformed includes the ENVELOPEs and coordinates listed under either POINT or COORDS.

FILTERCOORDSYS is used to specify the coordinate system of the requesting client. FEATURECOORDSYS is used to specify the coordinate system to which the service should be transformed. For more information on these elements, see Using Projection Elements.

In this first example, the client is in Robinson, so FILTERCOORDSYS *id="54030"*. In the request, the coordinates in the SPATIALFILTER are in Robinson. FEATURECOORDSYS also has an *id="54030"*, which means that the requested data should be returned in Robinson. The request is made on the Cities layer (*id="4"*).

The service, by default, is in geographic coordinates. This means that the ArcIMS Spatial Server receives the request and first transforms the Robinson coordinates into geographic coordinates. After the extraction has been made, the coordinates are transformed back into Robinson coordinates. Depending on the complexity of the data and the amount of features extracted, the processing time can be considerable when data transformations are involved.

Using FILTERCOORDSYS and FEATURECOORDSYS in a GET_FEATURES request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES featurelimit="25" beginrecord="0" outputmode="xml"
geometry="true" envelope="true" compact="true">
      <LAYER id="4" />
      <SPATIALQUERY subfields="#ALL#" where="POPULATION > 10000000" >
        <FILTERCOORDSYS id="54030" />
        <FEATURECOORDSYS id="54030" />
        <SPATIALFILTER relation="area_intersection">
          <ENVELOPE minx="-11395772" miny="930558" maxx="-3878142"
maxy="6419621"/>
        </SPATIALFILTER>
      </SPATIALQUERY>
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

In the response, the coordinates inside the ENVELOPE and COORDS elements are returned in Robinson.

FEATURES response with FILTERCOORDSYS and FEATURECOORDSYS set to Robinson ("54030"):

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <FEATURES>
      <FEATURE>
        <ENVELOPE minx="-9197766.55659269" miny="2075483.92733491"
maxx="-9197766.55659269" maxy="2075483.92733491"/>
        <FIELDS NAME="Mexico City" COUNTRY="Mexico"
POPULATION="14100000" CAPITAL="Y" #SHAPE#="[Geometry]" #ID#="221" />
        <MULTIPOINT>
          <COORDS>-9197766.55659269 2075483.92733491</COORDS>
        </MULTIPOINT>
      </FEATURE>
      <FEATURE>
        <ENVELOPE minx="-6419931.44911452" miny="4351163.61842219"
maxx="-6419931.44911452" maxy="4351163.61842219"/>
        <FIELDS NAME="New York" COUNTRY="US" POPULATION="16472000"
CAPITAL="N" #SHAPE#="[Geometry]" #ID#="549" />
        <MULTIPOINT>
          <COORDS>-6419931.44911452 4351163.61842219</COORDS>
        </MULTIPOINT>
      </FEATURE>
      <FEATURECOUNT count="2" hasmore="false" />
    </FEATURES>
  </RESPONSE>
</ARCXML>
```

In the next example, the client is once again in Robinson, and FILTERCOORDSYS *id="54030"*. However, the data to be returned is requested in geographic coordinates. Therefore, FEATURECOORDSYS is set to *id="4326"*. This request should process faster than the previous one. When the ArcIMS Spatial Server receives the request, the Robinson coordinates must still be transformed to geographic coordinates. However, since the data is requested back in geographic coordinates, no further transformation is needed on the data.

Using FILTERCOORDSYS in Robinson ("54030") and FEATURECOORDSYS in geographic coordinates ("4326") in a GET_FEATURES request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES featurelimit="25" beginrecord="0" outputmode="xml"
geometry="true" envelope="true" compact="true">
      <LAYER id="4" />
      <SPATIALQUERY subfields="#ALL#" where="POPULATION > 10000000" >
        <FILTERCOORDSYS id="54030" />
        <FEATURECOORDSYS id="4326" />
        <SPATIALFILTER relation="area_intersection">
```

```
          <ENVELOPE minx="-11395772" miny="930558" maxx="-3878142"
maxy="6419621"/>
        </SPATIALFILTER>
      </SPATIALQUERY>
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

In the response, the coordinate data in the ENVELOPE and COORDS elements is in geographic coordinates.

FEATURES response with FILTERCOORDSYS set to Robinson (54030) and FEATURECOORDSYS set to geographic coordinates (4326):

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <FEATURES>
      <FEATURE>
        <ENVELOPE minx="-99.127571105957" miny="19.4270458221436"
maxx="-99.127571105957" maxy="19.4270458221436"/>
        <FIELDS NAME="Mexico City" COUNTRY="Mexico"
POPULATION="14100000" CAPITAL="Y" #SHAPE#="[Geometry]" #ID#="221" />
        <MULTIPOINT>
           <COORDS>-99.127571105957 19.4270458221436</COORDS>
        </MULTIPOINT>
      </FEATURE>
      <FEATURE>
        <ENVELOPE minx="-74.0999984741211" miny="40.75" maxx="-
74.0999984741211" maxy="40.75"/>
        <FIELDS NAME="New York" COUNTRY="US" POPULATION="16472000"
CAPITAL="N" #SHAPE#="[Geometry]" #ID#="549" />
        <MULTIPOINT>
          <COORDS>-74.0999984741211 40.75</COORDS>
        </MULTIPOINT>
      </FEATURE>
      <FEATURECOUNT count="2" hasmore="false" />
    </FEATURES>
  </RESPONSE>
</ARCXML>
```

Summary of Geocoding Elements

Introduction

Geocoding is the process of identifying the coordinates of a location given its address. ArcXML uses geocode elements in four situations:

- To specify a layer's geocoding properties in a map configuration file.
- To request a point based on an address.
- To receive a response about geocoding information in the map configuration file.
- To receive a response with the location of an address.

The following table lists the nine ArcXML geocode elements and where each element is used:

Geocode Element	Used in:
ADDRESS	GET_GEOCODE request.
FEATURECOORDSYS	GET_GEOCODE request.
GCCOUNT	GEOCODE response.
GCFIELD	Map configuration file.
GCINPUT	SERVICEINFO response.
GCSTYLE	Map configuration file and SERVICEINFO response.
GCTAG	GET_GEOCODE request.
GEOCODE	GEOCODE response.
GET_GEOCODE	GET_GEOCODE request.

GET_GEOCODE and GEOCODE are valid only with Image and Feature Services. They are not valid with ArcMap Image Services.

Geocode Elements Used in a Map Configuration File

When creating a geocoded layer in a map configuration file, the information for geocoding is included in the EXTENSION element. The extension identifies the style of geocoding that should be used. There are two elements used inside EXTENSION:

- GCFIELD
- GCSTYLE

GCSTYLE is used to list the address style. GCFIELD is used to describe the different input parameters from the database needed for that style. Below is an example of using GCSTYLE and GCFIELD in an EXTENSION.

Using GCSYTLE and GCFIELD in an EXTENSION:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
       <ENVELOPE minx="-71.077092" miny="42.357962" maxx="-71.034511"
maxy="42.385263" name="Initial_Extent" />
       <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
       <SHAPEWORKSPACE name="shp_ws-64" directory="<path to data>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="Streets" visible="true" id="4">
       <DATASET name="bosstreets" type="line" workspace="shp_ws-64" />
       <SIMPLERENDERER>
         <SIMPLELINESYMBOL type="solid" width="2" color="255,0,0" />
       </SIMPLERENDERER>
       <EXTENSION type="Geocode">
         <GCSTYLE name="USAddressZ">
<!--reqd--> <GCFIELD id="FromLeft" name="L_F_ADD" />
<!--reqd--> <GCFIELD id="FromRight" name="R_F_ADD" />
<!--reqd--> <GCFIELD id="ToLeft" name="L_T_ADD" />
<!--reqd--> <GCFIELD id="ToRight" name="R_T_ADD" />
            <GCFIELD id="PreDir" name="PREFIX" />
            <GCFIELD id="PreType" name="PRE_TYPE" />
<!--reqd--> <GCFIELD id="StreetName" name="NAME" />
            <GCFIELD id="StreetType" name="TYPE" />
            <GCFIELD id="SufDir" name="SUFFIX" />
<!--reqd--> <GCFIELD id="LeftZone" name="ZIPL" />
<!--reqd--> <GCFIELD id="RightZone" name="ZIPR" />
         </GCSTYLE>
       </EXTENSION>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

ArcIMS has 10 standard address styles, most of which are valid only in the United States. See the Notes and Examples Sections for GCSTYLE and *Using ArcIMS*, the "Setting Geocode Properties" section of Chapter 3, for more information on address styles. ArcIMS also supports the implementation of custom address styles. See Using Custom Address Styles for information on customizing address styles for use in another country or for special cases in the United States.

Geocode Elements Used in GET_GEOCODE Requests

All geocode requests must be routed to the Geocode Server. This routing information is contained in the URL that is sent to the ArcIMS site such as in the following example (all one line):

http://myComputer.domain.com/servlet/com.esri.esrimap.Esrimap?ClientVersion=3.1
&ServiceName=myservice
&CustomService=Geocode
&Form=True&Encode=True

Requests to geocode an address are made using GET_GEOCODE. The geocode elements used in this request are:

- GET_GEOCODE
- FEATURECOORDSYS
- ADDRESS
- GCTAG

The address to geocode is added to GCTAG in the request. The *id* and *value* in GCTAG vary depending on which address style is used. In the example below, the "STREET" *id* is "380 New York Street" and the "Zone" *id* is "92373". No data is added to "CrossStreet".

A GET_GEOCODE request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_GEOCODE maxcandidates="25" minscore="60">
      <LAYER id="4" />
      <ADDRESS>
        <GCTAG id="STREET" value="380 New York Street" />
        <GCTAG id="Zone" value="92373" />
        <GCTAG id="CrossStreet" value="" />
      </ADDRESS>
    </GET_GEOCODE>
    </REQUEST>
</ARCXML>
```

FEATURECOORDSYS is needed if the x,y cooordinates for the geocoded point are in a different coordinate system than the ArcIMS service. For example, if the service is in geographic coordinates, but the client is in Robinson, then FEATURECOORDSYS should be used to request the coordinates in Robinson. The next example requests the points in Robinson (*id="53030"*). For more information on using FEATURECOORDSYS, see Using Projection Elements.

A GET_GEOCODE request with FEATURECOORDSYS:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_GEOCODE maxcandidates="25" minscore="60">
      <LAYER id="4" />
      <FEATURECOORDSYS id="53030">
      <ADDRESS>
        <GCTAG id="STREET" value="380 New York Street" />
        <GCTAG id="Zone" value="92373" />
```

```
        <GCTAG id="CrossStreet" value="" />
      </ADDRES>
    </GET_GEOCODE>
    </REQUEST>
</ARCXML>
```

Geocode Elements Used in a GEOCODE Response

A GEOCODE response is returned based on information in the GET_GEOCODE request. Geocode elements included in the response are:

- GEOCODE
- GCCOUNT

For each candidate returned in the response, address information and its location are included. The total number of returned features is included in GCCOUNT. An example GEOCODE follows.

A GEOCODE response:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
   <RESPONSE>
     <GEOCODE>
       <FEATURE featureid="1">
         <FIELD type="4" name="SCORE" size="5" precision="0">
           <FIELDVALUE valuestring="100" />
         </FIELD>
         <FIELD type="12" name="ADDRESSFOUND" size="21" precision="0">
           <FIELDVALUE valuestring="380 NEW YORK ST 92373" />
         </FIELD>
         <FIELD type="-98" name="SHAPEFIELD">
           <FIELDVALUE>
         <POINT x="-117.19496116" y="34.05777355" />
           </FIELDVALUE>
         </FIELD>
       </FEATURE>
       <GCCOUNT count="1" />
     </GEOCODE>
   </RESPONSE>
</ARCXML>
```

Geocode Elements Used in a SERVICEINFO Response

If an ArcIMS service includes a geocodable layer, the extension information is included in a SERVICEINFO response. To get this information, set the attribute *extensions* to "true" in the GET_SERVICE_INFO request. An example request follows.

GET_SERVICE_INFO request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_SERVICE_INFO extensions="true" renderer="false" fields="false"
envelope="false" />
  </REQUEST>
</ARCXML>
```

The request can be sent directly to the Image or Feature Server or can be routed to the Geocode Server. The information returned is different depending on which server is used. If the request is sent to the Image or Feature Server, the extension information is returned but includes only GCSTYLE.

SERVICEINFO response with ArcIMS service EXTENSION information and GCSYTLE:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <IMAGELIMIT pixelcount="2250000" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-71.0718204242754" miny="42.368904975182"
maxx="-71.0475995680561" maxy="42.3869647980717" name="Initial_Extent"
/>
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="bosstreets"
id="1" maxscale="0.00004703130261735 83">
        <FCLASS type="line"></FCLASS>
        <EXTENSION type="Geocode" >
          <GCSTYLE name="USAddressZ" />
        </EXTENSION>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

If the GET_SERVICE_INFO request is routed to the Geocode Server, the SERVICEINFO response includes the following geocode elements:

- GCSTYLE
- GCINPUT

A sample response is:

SERVICEINFO response rerouted to the Geocode Server:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
         <SEPARATORS cs=" " ts=";"/>
      </ENVIRONMENT>
      <LAYERINFO name="streets" id="1" >
         <EXTENSION type="geocode">
           <GCSTYLE name="USAddressZ" >
            <GCINPUT id="STREET" type="text" label="Street" width="10"
description="street number, street name and type" />
            <GCINPUT id="ZONE" type="text" label="Zone" width="5"
description="zone information" />
            <GCINPUT id="CROSSSTREET" type="text" label="Cross street"
width="10" description="cross street name and type" />
           </GCSTYLE>
         </EXTENSION>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

The *id* attribute values for GCINPUT are the same values that need to be included in GCTAG in the GET_GEOCODE request.

Using Custom Address Styles

Introduction

Note: Detailed descriptions of the standard geocoding rules and general geocoding principles can be found in the *Geocoding Developer's Kit* available from ESRI.

Geocoding is the process of identifying the coordinates of a location given its address. This document explains how to implement a custom address style in ArcIMS and provides a reference to the ArcXML elements that compose an ArcIMS style file.

This document assumes an understanding of geocoding; a working knowledge of ArcIMS, specifically how to publish ArcIMS services; and experience with creating custom address styles.

In this document, you are introduced to:

- Steps to implement custom address styles into ArcIMS
- The format and elements of the ArcIMS style file
- An example of an ArcIMS style file

What Are Address Styles?

ArcIMS has 10 standard address styles. Each address style consists of a list of address components that reference a layer's attribute table. Each style has optional and required address components. The required components need to be filled with a data field from the attribute table. Each address style contains a list of preferred field names for each component. Using the preferred field names, ArcIMS will determine a probable field from the attribute table for each component.

Each address style consists of a set of standard files such as *.stn, *.dct, *.cls, and *.mat. In addition to these files, ArcIMS uses additional style files with *.stl and *.st_ extensions. The ArcIMS style files are composed of XML-compliant elements written in ArcXML that contain geocoding settings specific to ArcIMS.

The address styles available with ArcIMS are:

- U.S. addresses without zone information address style (USAddress.stl)
- U.S. addresses with zone information address style (USAddressZ.stl)
- U.S. single house address style (USSingleHouse.stl)
- U.S. single house with zone address style (USSingleHouseZ.stl)
- U.S. single range address style (USSingleRange.stl)
- U.S. single range with zone address style (USSingleRangeZ.stl)
- ZIP+4 address style (Zip4.stl)
- ZIP+4 range address style (Zip4Range.stl)

- 5-digit ZIP address style (Zip5.stl)
- Single field address style (SingleField.stl)

If the standard address styles do not meet your needs, the following pages provide the steps for integrating a new style file into ArcIMS. The reference elements and example can be used as a guide for creating the *.stl and *.st_ files.

Implementing a Custom Address Style

These steps assume:

- The standard address style files for the custom style have been created. Refer to the Geocoding Developer's Kit for information regarding the files that compose an address style.
- The ArcIMS style files (*.stl and *.st_) have been created. The *.st_ file is necessary if you want the custom style to support street intersection geocoding. You can create a *.st_ file or, depending on your needs, use the ArcIMS USStreetInts.st_ file.

The following pages offer a complete reference to the elements of the *.stl file. The final pages offer an example of a *.stl file as implemented for address matching using a German dataset.

1. Copy all of the address style files to the \Server\ext\GeocodeServer\Styles and \IndexBuilder\Styles directories. If you have already implemented the custom address style in ArcView 3.x, you can copy all the files from the \geocode directory into the two ArcIMS Styles folders.
2. Copy the ArcIMS style file (*.stl and *.st_) to the \Server\ext\GeocodeServer\Styles and \IndexBuilder\Styles directories.
3. Locate the esri_mo10res.jar file. The ArcIMS installation puts the esri_mo10res.jar file in the JRE's lib\ext directory. The default location for the JRE is C:\Program Files\JavaSoft\JRE\1.3\lib\ext.
4. Unzip the contents of the esri_mo10res.jar file to an empty directory. In this example, c:\temp is used. After extracting, you will have two subdirectories: com and meta-inf.
5. Create a properties file for the custom address style. The naming convention for the file should be .properties, for example, Germanaddress.properties.

   Navigate to the com\esri\mo\res\props directory. This directory contains property files for the styles already supported by ArcIMS such as USAddressZ.properties and Zip4.properties.

   To create the new properties file:
   - Get the information for the content from the ArcIMS style file.
   - Use the available property files as a guideline for format of the content.
6. Open the c:\temp\com\esri\mo\res\props\GeoCodeStyles.properties file in a text editor and add a listing for your address style. The following shows what the

GeoCodeStyles.properties file looks like with additions for German Addresses and German Addresses with Zone address styles.

```
# GeoCodeStyles.properties
# entries must have the form
# N: Property name, Property label, PropertyResourcePathname
# where "N" must be an integer, and must be unique across
entries, but can be in any order.
# "Property name" is a simple name that must not contain blanks
or ","
# "Property label" is a user interface label
# "PropertyResourcePathname" is a fully-qualified resource name
that must not contain a ","
1: USAddressZ, US Streets with Zone,
/com/esri/aims/resources/sdk/propsUSAddressZ.properties
2: USAddress, US Streets,
/com/esri/aims/resources/sdk/propsUSAddress.properties
9: Zip5, ZIP 5-Digit,
/com/esri/aims/resources/sdk/propsZip5.properties
10: SingleField, Single Field,
/com/esri/aims/resources/sdk/propsSingleField.properties
6: USSingleHouse, US Single House,
/com/esri/aims/resources/sdk/propsUSSingleHouse.properties
5: USSingleHouseZ, US Single House with Zone,
/com/esri/aims/resources/sdk/propsUSSingleHouseZ.properties
4: USSingleRange, US Single Range,
/com/esri/aims/resources/sdk/propsUSSingleRange.properties
3: USSingleRangeZ, US Single Range with Zone,
/com/esri/aims/resources/sdk/propsUSSingleRangeZ.properties
7: Zip4, ZIP+4, /com/esri/aims/resources/sdk/propsZip4.properties
8: Zip4Range, ZIP+4 Range,
/com/esri/aims/resources/sdk/propsZip4Range.properties
11: GermanAddress, German Addresses,
/com/esri/aims/resources/sdk/propsGermanAddress.properties
12: GermanAddressZ, German Addresses with Zone,
/com/esri/aims/resources/sdk/propsGermanAddressZ.properties
```

7. Zip the contents of the c:\temp directory into a new esri_mo10res.jar file. Remove the original version of esri_mo10res.jar from the JRE's lib\ext directory and copy the new version to this location.
8. Stop and restart ArcIMS by following these steps: Stop ArcIMS Monitor and then stop ArcIMS Application Server. Start ArcIMS Application Server and start ArcIMS Monitor.

You can now use this address style when setting geocoding properties in ArcIMS Author. See chapter 3 in *Using ArcIMS* for details on how to set geocoding properties and build the geocoding index.

Elements of the ArcIMS Style File

The ArcIMS style file (*.stl) is composed of XML-compliant elements written in ArcXML for geocoding settings specific to ArcIMS.

An ArcIMS style file contains a description of a single ArcIMS address style. The style depends on the address style's geocoding rules and contains some information taken from *.dct, *.mat, and *.stn files. In addition to the ArcIMS style file, a *.st_ file is needed for ArcIMS styles that support geocoding of street intersections.

The elements of the style file are described on the following pages. The examples use the USAddressZ.stl and USStreetInts.st_ files. The elements are listed in the order in which they appear in the style file.

GEOSTYLE

Purpose: Root element for the *.stl file.

The two required attributes are *name* and *description*.

The value of the name attribute is the name of the *.stl file. The value of the description attribute is a text string describing the address style.

```
<GEOSTYLE name="USAddressZ" description="US addresses with zone
information" >...</GEOSTYLE>
```

STNCOMMAND

Purpose: Defines the *.stn file used for the style.

The one required attribute is *file*.

The value of the file attribute is the name of the *.stn file.

```
<STNCOMMAND file="us_addr.stn" />
```

MATCHKEY

Purpose: Defines the *.dct file used for the style.

The one required attribute is *file*.

The value of the file attribute is the name of the *.dct file.

```
<MATCHKEY file="us_addr.dct" />
```

MATCHRULES, MATCH

MATCHRULES

Purpose: Defines the *.mat file used for the style.

MATCH

Purpose: Contains a list of matched (*mprop*) and unmatched (*uprop*) probabilities defined in the *.mat file.

```
<MATCHRULES file="us_addr1.mat" >
  <MATCH mprob="0.9" uprob="0.01" />
  <MATCH mprob="0.9" uprob="0.01" />
  <MATCH mprob="0.8" uprob="0.1" />
  <MATCH mprob="0.7" uprob="0.1" />
  <MATCH mprob="0.85" uprob="0.1" />
  <MATCH mprob="0.85" uprob="0.1" />
  <MATCH mprob="0.999" uprob="0.05" />
</MATCHRULES>
```

QUERY

Purpose: Declares the format of the expression used to get potential candidates from the database. The expression is a query that is performed on indexes built for the data by the IndexBuilder.

The *stn* and *zip* attributes are index names defined by the INDEXES element (see below). The values of the attributes are names of proper match key fields from the *.dct file defined by the MATCHKEY element. The Geocode Server takes the contents of the fields to form the expression.

```
<QUERY stn="SN" zone="ZN" />
```

For example, the input address is "380 New York st 92373". After standardization, the "SN" field of the match key contains "New York" and the "ZN" field contains "92373".

Using the QUERY shown above, the Geocode Server generates the following expression:

stn="New York" & zone="92373"

The indexing functionality of the Geocode Server performs this query and gets all records with field "StreetName" equals "New York" and field "LeftZone" equals "92373" or field "RightZone" equals "92373".

INPUT, TAG, FORMAT, STNSTRING, ADDTAG, MATCHKEYFIELD

INPUT

Purpose: Defines the format of the request sent from the client to the Geocode Server.

The client sends a request to the Geocode Server. The request is parsed and standardized and sent back to the client. In ArcIMS, the client can be ArcExplorer 4, the HTML Viewer, or the Java Standard or Java Custom Viewers.

```
<INPUT>
  <TAG id="STREET" label="Street" width="10" type="text"
description="street number, street name and type" />
  <TAG id="zone" label="zone" width="5" type="text" description="zne
information (5 digits)" />
```

```
  <FORMAT>
    <STNSTRING>
      <ADDTAG>STREET</ADDTAG>
    </STNSTRING>
    <MATCHKEYFIELD zn="zone" />
  </FORMAT>
</INPUT>
```

TAG

Purpose: Each TAG defines the format of a single input element that must be used in the request. The *id* attribute defines the unique identifier of this element. It is also used in the request. The *label*, *width*, *type*, and *description* attributes contain a description of the element and are used only by the client.

FORMAT

Purpose: Defines how information retrieved from the request is handled.

STNSTRING

Purpose: Defines a list of input elements whose values form the address string for standardization.

ADDTAG

Purpose: Defines the name of a single input element for the STNSTRING element.

MATCHKEYFIELD

Purpose: Defines a list of input elements whose values go directly into match key.

ArcIMS address styles require that addresses be sent in as separate pieces that can be processed in different ways before the whole address is passed for standardization. Each address needs to be standardized before it can be matched. See the *Geocoding Developer's Kit* for details.

An example of how an address is broken down is a U.S. address that is divided into two parts: STREET and ZONE (usually ZIP Codes). The STREET includes house number, street name, and street type. Only the STREET part needs to be parsed for standardization. The ZONE part is standardized and can be put directly into a match key. The STNSTRING element defines which parts need to be standardized, and the MATCHKEYFIELD element defines which parts can go directly into match key.

The STNSTRING element can have many ADDTAG elements. In this example the address can be divided into four parts: HouseNumber, StreetName, StreetType, and ZONE. The first three parts are concatenated to form the address string for standardization. The following shows what the INPUT element should look like:

```
<INPUT>
  <TAG id="HOUSENUMBER" ... />
  <TAG id="STREETNAME" ... />
  <TAG id="STREETTYPE" ... />
```

```
  <TAG id="zone" ... />
  <FORMAT>
    <STNSTRING>
      <ADDTAG>HOUSENUMBER</ADDTAG>
      <ADDTAG>STREETNAME</ADDTAG>
      <ADDTAG>STREETTYPE</ADDTAG>
    </STNSTRING>
    <MATCHKEYFIELD zn="zone" />
  </FORMAT>
</INPUT>
```

The MATCHKEYFIELD element may have more than one attribute. The name of each attribute is the name of a match key's field defined in the *.dct file. The value of an attribute is equal to the value of *id* of the corresponding TAG.

In the example above, the MATCHKEYFIELD element is defined as:

```
<MATCHKEYFIELD zn="zone" />
```

The MATCHKEYFIELD element can be defined with more than one attribute, for example:

```
<MATCHKEYFIELD zn="zone" ct="city" st="state" />
```

The following example shows a request that can be used for a layer associated with the USAddressZ.stl style:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_GEOCODE maxcandidates="25"  minscore="60">
      <LAYER id="streets" />
      <ADDRESS>
        <GCTAG id="STREET" value="380 New York St" />
        <GCTAG id="Zone" value="92373" />
        <GCTAG id="CrossStreet" value="" />
      </ADDRESS>
    </GET_GEOCODE>
  </REQUEST>
</ARCXML>
```

In this example, <GCTAG id="STREET"../> corresponds to INPUT's <TAG id="STREET" .../>, and <GCTAG id="zone" .../> corresponds to <TAG id="zone".../>.

STREET1, STREET2

STREET1

Purpose: Used in INPUT for street intersection style.

STREET2

Purpose: Used in INPUT for street intersection style.

The request in the above example contains three input GCTAG elements, while USAddressZ.stl defines only two. This is because this style may also be used for street intersection geocoding. The third element, CrossStreet, is defined in a street intersection style file, USStreetInts.st_.

The format of the *.st_ file is slightly different from the *.stl file format. The INPUT element for the *.st_ file looks like the following:

```
<INPUT>
  <STREET1 id="STREET" />
  <STREET2 id="CROSSSTREET" label="Cross street" width="10" type="text"
description="cross street name and type" />
</INPUT>
```

See the USStreetInts.st_ for more details.

The INPUT element from the *.st_ file always has two predefined child elements: STREET1 and STREET2.

The STREET1 element defines the TAG from the corresponding *.stl file, which is always a "street tag". The STREET2 element defines an additional input element, which will be used to specify the second street for street intersection geocoding. This INPUT element does not need a FORMAT child element because ZIP code is not used for street intersection geocoding.

OUTPUT, PRINTMATCHKEY, PRINTFIELD, SEPARATOR

OUTPUT

Purpose: Defines the format of the response to the client from the Geocode Server.

The client sends a request to the Geocode Server. This request is parsed and standardized, stored in the MATCHKEYFIELD, and sent back to the client.

In this next example, you can see that the MATCHKEYFIELD is printed as part of the output.

```
<OUTPUT>
  <PRINTMATCHKEY>HN</PRINTMATCHKEY>
  <PRINTFIELD>PreDir</PRINTFIELD>
  <PRINTFIELD>PreType</PRINTFIELD>
  <PRINTFIELD>StreetName</PRINTFIELD>
  <PRINTFIELD>StreetType</PRINTFIELD>
  <PRINTFIELD>SufDir</PRINTFIELD>
  <PRINTMATCHKEY>ZN</PRINTMATCHKEY>
</OUTPUT>
```

PRINTMATCHKEY
Purpose: Used in OUTPUT to define output format.

PRINTFIELD
Purpose: Used in OUTPUT to define output format.

SEPARATOR
Purpose: Used in OUTPUT to define output format.

After geocoding, the match candidates are sent back to the client. Each candidate has a:

- **score** - reflects the accuracy of the candidate
- **pinpoint**- location of the candidate on the map
- **address**- information found in the database

OUTPUT defines how the address string looks. The address string's format is defined by:

- The names of the fields for the different parts of the address
- The order in which the parts are concatenated

It may be useful to get some of the parts of the address directly from match key such as house number or ZIP Code.

PRINTFIELD defines the database fields (see FIELDS below). PRINTMATCHKEY defines the match key fields. The order in which these elements are inserted into the OUTPUT element is important because it defines how all the fields will be combined.

If a field listed in OUTPUT is empty, the field is not included in the address string. The next example shows how the OUTPUT element works for the string:

380 NEW YORK ST 92373

- 380 - contents of match key field HN
- <empty> - contents of field with id="PreDir"
- <empty> - contents of field with id="PreType"
- NEW YORK - contents of field with id="StreetName"
- ST - contents of field with id="StreetType"
- <empty> - contents of field with id="SufDir"
- 92373 - contents of match key field ZN

The OUTPUT element can also include SEPARATOR to define a character to insert between the fields. The next example shows OUTPUT from Zip4.stl:

```
<OUTPUT>
<PRINTMATCHKEY>ZP</PRINTMATCHKEY>
<SEPARATOR>-</SEPARATOR>
<PRINTMATCHKEY>Z4</PRINTMATCHKEY>
```

```
</OUTPUT>
```

In this case the output string looks like this: 12345-1234.

The OUTPUT element used in the *.st_ file has a different format than the OUTPUT element for the *.stl file. The result of geocoding a street intersection is a point where two streets intersect. The address string contains information about both streets, usually the street names and types. The OUTPUT element is used to define where to put the delimiter between the first and second streets:

```
<OUTPUT separator="+" position="42" />
```

The *separator* attribute defines a symbol to be used as a delimiter. The value of the *separator* can be any XML-compliant character.

The *position* attribute defines where the delimiter is placed. The *position* value defines the length of the first street's information in the candidate's buffer generated by the Geocode Server. The value depends on how the streets are defined in the corresponding *.mat file. For example, in the us_intsc.mat file the first street information is defined to be 42 characters long.

The table below shows that the first street information includes street name (StreetName1), street type (Type1), and suffix direction (SufDir1). The suffix direction begins at the forty-first position and is two characters long, thus making the first street information 42 characters long.

VAR PreDir1	1	2	X ; Prefix direction 1
VAR PreType1	3	4	X ; Prefix street type 1
VAR StreetName1	7	28	S ; Street name 1
VAR Type1	35	6	X ; Suffix street type 1
VAR PreDir2	43	2	X ; Prefix direction 1
VAR SufDir1	41	2	X ; Suffix direction 1
VAR PreDir1	1	2	X ; Prefix direction 2
VAR PreType2	45	4	X ; Prefix street type 1
VAR PreDir1	1	2	X ; Prefix street type 2
VAR StreetName2	49	28	S ; Street name 2
VAR Type2	77	6	X ; Suffix street type 2
VAR SufDir2	83	2	X ; Suffix direction 2

The USStreetInts.st_ file, which is associated with the us_intsc.mat file, uses position="42". An example address string is "NEW YORK ST + STATE ST".

See the INTERSECTION, INTERSECTIONSTYLE, and INTERSECTIONQUERY elements for more information about geocoding with street intersections.

FIELDS, FIELD, POSITION, PREFERREDNAMES, NAME

FIELDS

Purpose: Defines all database fields used for geocoding.

```
<FIELDS>
  lt;FIELD id="FromLeft" required="true" description="Left from house
number">
    <POSITION start="1" width="10" />
    <PREFERREDNAMES>
      <NAME>L-F-ADD</NAME>
      <NAME>L_F_ADD</NAME>
      . . . . . .
      <NAME>LEFTADD1</NAME>
    </PREFERREDNAMES>
  </FIELD>
  . . . . . .
</FIELDS>
```

All the information used for the FIELDS element is taken from the corresponding *.mat file and addstyle.db file.

FIELD

Purpose: Defines the single field used in FIELDS.

Each FIELD element must have the following attributes:

- *id*- Value is the name of the VAR variable copied from *.mat file. It is used everywhere in the *.stl file to refer to this particular field. This is not a field name in a database; the name in the database is unknown until the style is linked with an actual database when the map configuration file is started as an ArcIMS service.
- *required*- Value should be "true" if the field is required or "false" for optional fields. All required fields must be used in the map configuration file.
- *description*-Value is a text string describing the field. This description can be copied from *.mat.

POSITION

Purpose: Defines the position of the field in the candidate's buffer (the same as in *.mat file).

PREFERREDNAMES

Purpose: Defines the list of preferred names for fields.

NAME

Purpose: Defines the single name from the PREFERREDNAMES element.

POSITION defines the position of the field in an internal buffer. PREFERREDNAMES is a list of the field's preferred names in a database. The *start* and *width* attributes are

copied from the *.mat file. The list of preferred names is copied from addstyle.db.

The format of the FIELDS element in the *.st_ file is different from those used in the *.stl file:

```
<FIELDS>
  lt;FIELD id="PreDir1" required="false" datafieldid="PreDir"
belongsto="street1">
    <POSITION start="1" width="2" />
  </FIELD>
  . . . . . .
  <FIELD id="PreDir2" required="false" datafieldid="PreDir"
belongsto="street2">
    <POSITION start="43" width="2" />
  </FIELD>
  . . . . . .
</FIELDS>
```

The FIELDS element also reflects the information from the corresponding *.mat file. Each FIELD element must have an *id* copied from the VAR variable. The required attribute must have a value of "true" for required fields and a value of "false" for the optional fields. The *datafield* attribute links the field with the FIELD defined in the corresponding *.stl file. The *belongsto* attribute defines whether the field belongs to STREET1 or STREET2.

FIELDS from the street intersection style do not need PREFERREDNAMES; this information will be taken from the *.stl file.

INDEXES, INDEX

INDEXES

Purpose: Defines the geocoding indexes used for ArcIMS address styles.

```
<INDEXES>
<INDEX name="STN" seqnumber="1" hash="soundex" duplicate="true">
<FIELD id="StreetName" />
</INDEX>
<INDEX name="zone" seqnumber="2" hash="random" duplicate="true">
<FIELD id="LeftZone" />
<FIELD id="RightZone" />
</INDEX>
</INDEXES>
```

Information defined here is needed for the IndexBuilder to build the indexes and is also used by the Geocode Server to retrieve candidates from the database.

INDEX

Purpose: Defines a single geocoding index. As many as 255 indexes may be defined.

Each INDEX must have the following attributes:

- *name* - Value is a unique name of the index; should be 1-8 characters beginning with a letter.
- *seqnumber* - Value is an integer representing a unique number for the index; the first index should be given a number of 1, etc.
- *hash* - Value specifies the type of index, either "random" or "soundex". Use "soundex" to detect spelling variation of the key (for example, street names). Use "random" when spelling errors are not applicable (for example, ZIP Code).
- *duplicate* - Value of "true" allows duplicate keys, and "false" does not allow duplicates. If duplicates are allowed, there can be multiple records for one key. For example, a street name such as "Main" may point to a number of records in a database.

FIELD defines which field from the database will be indexed. The *id* attribute refers to a FIELD defined in FIELDS.

INDEX may have two FIELD child elements. In this example, a combined index is created, such an index contains all different keys from both specified fields. For example, in the code shown above, index zone includes two FIELD *id* values - LeftZone and RightZone. It means that when the index is built it will get only one key for a record where LeftZone and RightZone fields have the same value and two keys for a record where the fields have different values.

INTERSECTION

Purpose: Defines a street intersection style file.

```
<INTERSECTION style="USStreetInts.st_" />
```

The style attribute refers to the *.st_ file. This style defines rules for street intersection geocoding and can be "linked" with other street styles such as USAddress or USSingleHouse. It cannot be used by itself (this is why it has the *.st_ extension) or with other nonstreet styles such as Zip4 or SingleField.

See the SEPARATOR element for more information about the separator that is used for the OUTPUT of geocoding street intersections.

INTERSECTIONSTYLE

Purpose: Root element for the street intersection style. This element has no attributes.

```
<INTERSECTIONSTYLE>...</INTERSECTIONSTYLE>
```

INTERSECTIONQUERY

Purpose: Defines index expression used to extract potential candidates from the database. This element has the same purpose as QUERY element for the *.stl files.

```
<INTERSECTIONQUERY indexname="stn" mkfield1="S1" mkfield2="S2"
indexname="zone" mkefieldzone="ZN" />
```

The *indexname* attribute defines the name of the index (declared in the INDEXES element in the *.stl file) used for street intersection geocoding. The *mkfield1* and *mkfield2* attributes define names of match key's fields (from the *.dct).

The *indexnamezone* attribute defines the name of the index (declared in the INDEXES element in the *.stl file) used for street intersection with zone geocoding.

The *mkfield1* and *mkfield2* attributes define names of match key's fields (from the *.dct file).

The *mkfieldzone* attribute defines the name of the zone match key field (from the *.dct file).

In order to get potential candidates for street intersections, the Geocode Server needs to perform the same query twice - for data from *mkfield1* and *mkfieldzone* and for data from *mkfield2* and *mkfieldzone*. Here is an example of two expressions that would be performed to find the intersection "New York st, 92373" and "State st, 92373":

First index expression:	stn="New York"? & zn="92373"
Second index expression:	stn="State"? & zn="92373"

All the records found by these two queries are then "matched" to find which candidates intersect. Note: The ? is used for the soundex indexing.

When creating a custom address style that uses intersection with zone geocoding, the *mkfieldzone* attribute's value should be "ZN".

Example ArcIMS Style File

The following is an example of an ArcIMS style file (*.stl) as implemented for German address matching.

Sample Style File for German Address Matching

```
<GEOSTYLE name=“GermanAddress” description=“German addresses without
zone information” >
<STNCOMMAND file=“ger_add.stn” />
<MATCHKEY file=“ger_add.dct” />
<MATCHRULES file=“ger_add2.mat” >
  <MATCH mprob=“0.9" uprob=”0.01" />
  <MATCH mprob=“0.7" uprob=”0.1" />
  <MATCH mprob=“0.999" uprob=”0.05" />
</MATCHRULES>
<QUERY stn=“SN” />
<INPUT>
  <TAG id=“STREET” label=“Street” width=“10" type=“text”
description=”street name, street suffix and street number” />
```

```
  <FORMAT>
    <STNSTRING>
         <ADDTAG>STREET</ADDTAG>
    </STNSTRING>
  </FORMAT>
</INPUT>
<OUTPUT>
  <PRINTFIELD>StreetName</PRINTFIELD>
  <PRINTFIELD>StreetSuffix</PRINTFIELD>
  <PRINTMATCHKEY>HN</PRINTMATCHKEY>
</OUTPUT>
<FIELDS>
  <FIELD id="StreetName" required="true" description="Street name">
    <POSITION start="1" width="32" />
    <PREFERREDNAMES>
      <NAME>STRAßEN.NAME</NAME>
      <NAME>STRAßEN_NAM</NAME>
      <NAME>STRAßEN-NAM</NAME>
      <NAME>STR.NAME</NAME>
      <NAME>STR_NAME</NAME>
      <NAME>STR-NAME</NAME>
      <NAME>STR.NAM</NAME>
      <NAME>STR_NAM</NAME>
      <NAME>STR-NAM</NAME>
      <NAME>ST.NAME</NAME>
      <NAME>ST_NAME</NAME>
      <NAME>ST-NAME</NAME>
      <NAME>NAME</NAME>
      <NAME>FNAME_BASE</NAME>
    </PREFERREDNAMES>
  </FIELD>
  <FIELD id="StreetSuffix" required="false" description="Suffix street
type">
    <POSITION start="33" width="16" />
    <PREFERREDNAMES>
      <NAME>STRAßEN.EXT</NAME>
      <NAME>STRAßEN_EXT</NAME>
      <NAME>STRAßEN-EXT</NAME>
      <NAME>STR.EXT</NAME>
      <NAME>STR_EXT</NAME>
      <NAME>STR-EXT</NAME>
      <NAME>ST.EXT</NAME>
      <NAME>ST_EXT</NAME>
      <NAME>ST-EXT</NAME>
    </PREFERREDNAMES>
  </FIELD>
  <FIELD id="FromLeft" required="true" description="Left from house
number">
    <POSITION start="49" width="8" />
    <PREFERREDNAMES>
      <NAME>L-ADR.VON</NAME>
      <NAME>L-ADR_VON</NAME>
      <NAME>L-ADR-VON</NAME>
      <NAME>L_ADR.VON</NAME>
      <NAME>L_ADR_VON</NAME>
      <NAME>L_ADR-VON</NAME>
      <NAME>LADR.VON</NAME>
```

```
      <NAME>LADR_VON</NAME>
      <NAME>LADR-VON</NAME>
      <NAME>L-V.ADR</NAME>
      <NAME>L-V_ADR</NAME>
      <NAME>L-V-ADR</NAME>
      <NAME>L_V.ADR</NAME>
      <NAME>L_V_ADR</NAME>
      <NAME>L_V-ADR</NAME>
      <NAME>LINKEADR1</NAME>
      <NAME>LREF_ADDRE</NAME>
    </PREFERREDNAMES>
  </FIELD>
  <FIELD id="FromRight" required="true" description="Right from house
number">
    <POSITION start="57" width="8" />
    <PREFERREDNAMES>
      <NAME>R-ADR.VON</NAME>
      <NAME>R-ADR_VON</NAME>
      <NAME>R-ADR-VON</NAME>
      <NAME>R_ADR.VON</NAME>
      <NAME>R_ADR_VON</NAME>
      <NAME>R_ADR-VON</NAME>
      <NAME>RADR.VON</NAME>
      <NAME>RADR_VON</NAME>
      <NAME>RADR-VON</NAME>
      <NAME>R-V.ADR</NAME>
      <NAME>R-V_ADR</NAME>
      <NAME>R-V-ADR</NAME>
      <NAME>R_V.ADR</NAME>
      <NAME>R_V_ADR</NAME>
      <NAME>R_V-ADR</NAME>
      <NAME>READR1</NAME>
      <NAME>RREF_ADDRE</NAME>
    </PREFERREDNAMES>
  </FIELD>
  <FIELD id="ToLeft" required="true" description="Left to house
number">
    <POSITION start="65" width="8" />
    <PREFERREDNAMES>
      <NAME>L-ADR.NACH</NAME>
      <NAME>L-ADR_NACH</NAME>
      <NAME>L-ADR-NACH</NAME>
      <NAME>L_ADR.NACH</NAME>
      <NAME>L_ADR_NACH</NAME>
      <NAME>L_ADR-NACH</NAME>
      <NAME>LADR.NACH</NAME>
      <NAME>LADR_NACH</NAME>
      <NAME>LADR-NACH</NAME>
      <NAME>L-N.ADR</NAME>
      <NAME>L-N_ADR</NAME>
      <NAME>L-N-ADR</NAME>
      <NAME>L_N.ADR</NAME>
      <NAME>L_N_ADR</NAME>
      <NAME>L_N-ADR</NAME>
      <NAME>LINKEADR2</NAME>
      <NAME>LNREF_ADDRE</NAME>
    </PREFERREDNAMES>
  </FIELD>
```

```
  <FIELD id=“ToRight” required=“true” description=“Right to house
number”>
    <POSITION start=“73" width=“8" />
    <PREFERREDNAMES>
      <NAME>R-ADR.NACH</NAME>
      <NAME>R-ADR_NACH</NAME>
      <NAME>R-ADR-NACH</NAME>
      <NAME>R_ADR.NACH</NAME>
      <NAME>R_ADR_NACH</NAME>
      <NAME>R_ADR-NACH</NAME>
      <NAME>RADR.NACH</NAME>
      <NAME>RADR_NACH</NAME>
      <NAME>RADR-NACH</NAME>
      <NAME>R-N.ADR</NAME>
      <NAME>R-N_ADR</NAME>
      <NAME>R-N-ADR</NAME>
      <NAME>R_N.ADR</NAME>
      <NAME>R_N_ADR</NAME>
      <NAME>R_N-ADR</NAME>
      <NAME>READR2</NAME>
      <NAME>RNREF_ADDRE</NAME>
    </PREFERREDNAMES>
  </FIELD>
</FIELDS>
<INDEXES>
  <INDEX name=“STN” seqnumber=“1" hash=“soundex” duplicate=“true”>
    <FIELD id=“StreetName” />
  </INDEX>
</INDEXES>
<INTERSECTION style=“GermanStreetInts.st_” />
</GEOSTYLE>
```

Section 2: Alphabetical Listing of ArcXML Elements and Attributes

ADD_RELEVANCE_FEEDBACK

Used in: REQUEST
Servers: Metadata (Browse)
Parent elements: GET_METADATA

<ADD_RELEVANCE_FEEDBACK

docid ="***string***"
relevance ="**1 - 10**"
>

No Child Elements
</ADD_RELEVANCE_FEEDBACK >

Bold: Attribute or child element is required.

Description:

Used to increment relevance values for each METADATA_DATASET returned in the response.

Restrictions:

None

Notes:

- See METADATA_ACTION for response.

Attribute Descriptions for ADD_RELEVANCE_FEEDBACK:

Attribute	Usage
docid	String used to uniquely identify a document. The client used to publish the metadata is responsible for creating the document ID. This ID is automatically assigned when using ArcCatalog. If another client is used, GET_UUID can be used to request a valid ID. The format for an ID is the following: {HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH} where H is a hexadecimal digit (0-9,a-f,A-F). The ID is limited to 38 characters.
relevance	Value between 1 and 10 indicating the relative relevance of the document to the client. The value is added to the existing value for a new relevance total.

Examples for ADD_RELEVANCE_FEEDBACK:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <ADD_RELEVANCE_FEEDBACK relevance="3" docid="{1F7DDF21-BC01-
4C20-8AA5-243B33ED0B1E}" />
    </GET_METADATA>
  </REQUEST>
</ARCXML>
```

ADDEDFEATURES

Used in: MARKUP
Parent elements: MARKUPLAYER

<ADDEDFEATURES >
No Attributes

(m) <FEATURE... />

</ADDEDFEATURES >

(m): Child element can be used multiple times.

Description:

Describes features in an XML file after features have been added to a layer using EditNotes.

Restrictions:

None

Notes:

None

Examples for ADDEDFEATURES:

Example 1:

```
<?xml version="1.0"?>
<MARKUP>
  <WORKSPACES>
    <FEATURESERVERWORKSPACE name="ifs_ws-0"
url="http://mycomputer.domain.com/servlet/com.esri.esrimap.Esrimap"
service="world" />
  </WORKSPACES>
  <MARKUPLAYER layername="Countries" workspace="ifs_ws-0">
    <ADDEDFEATURES>
      <FEATURE featureid="1000000">
      <ENVELOPE minx="-35.92572247220596" miny="-48.809104594884616"
maxx="-5.291989044369359" maxy="-21.047283675907693" />
      <FIELD name="AREA" precision="3" size="12" type="8" />
      <FIELD name="NAME" precision="0" size="40" type="12" />
      <FIELD name="ABBREVNAME" precision="0" size="12" type="12" />
      <FIELD name="FIPS_CODE" precision="0" size="2" type="12" />
      <FIELD name="WB_CNTRY" precision="0" size="3" type="12" />
      <FIELD name="#SHAPE#" precision="0" size="0" type="-98">
        <FIELDVALUE>
```

```
          <POLYGON>
            <RING>
            <POINT x="-19.17289950385782" y="-21.047283675907693" />
            <POINT x="-35.92572247220596" y="-38.2787587290658" />
            <POINT x="-10.078509892468816" y="-48.809104594884616" />
            <POINT x="-5.291989044369359" y="-24.397848269577324" />
            <POINT x="-19.17289950385782" y="-21.047283675907693" />
            </RING>
          </POLYGON>
        </FIELDVALUE>
      </FIELD>
      </FEATURE>
    </ADDEDFEATURES>
  </MARKUPLAYER>
</MARKUP>
```

ADDRESS

Used in: REQUEST
Servers: Geocode
Parent elements: GET_GEOCODE

<ADDRESS >
No Attributes

(m) <GCTAG... />

</ADDRESS >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Defines framework for geocoding an address in a GET_GEOCODE request.

Restrictions:

None

Notes:

- For more information on geocoding elements, see Summary of Geocoding Elements.

Examples for ADDRESS:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_GEOCODE maxcandidates="25"  minscore="60">
      <LAYER id="streets" />
      <ADDRESS>
        <GCTAG id="STREET" value="380 New York St" />
        <GCTAG id="Zone" value="92373" />
        <GCTAG id="CrossStreet" value="" />
      </ADDRESS>
    </GET_GEOCODE>
  </REQUEST>
</ARCXML>
```

ADMIN_TABLE

Used in: CONFIG
Parent elements: METADATA_CONFIG

<ADMIN_TABLE

idcolumn ="***string***"
insert ="***string***"
tablename ="***string***"
filter ="*string*"
>

(m) <RESPONSE_COLUMN... />

</ADMIN_TABLE >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Provides information on an external administration table.

Restrictions:

- The administration table is user-defined table used in addition to the metadata tables generated by the Metadata Server. You must create the table separately using your database administration tools. The table cannot be created using ArcCatalog.

Notes:

- The ADMIN_TABLE is an external database table joined to the main internal metadata table. The name of this internal table is defined in the configuration file under TABLE_NAME. The administration table is visible to an administrator, but not clients. The table can be used to store additional fields of data for use in queries.

Attribute Descriptions for ADMIN_TABLE:

Attribute	Usage
filter	SQL *where* clause to execute against the ADMIN_TABLE. The clause is appended with an AND to every *where* clause generated by SEARCH_METADATA.
idcolumn	Name of field in the ADMIN_TABLE containing the join time ussed to join the administration table to the main internal metadata table. The ID

	column in the ADMIN_TABLE must be a string field 38 characters in length.
insert	SQL INSERT statement to execute against the ADMIN_TABLE whenever a new document is added to the Metatdata Server. Must contain '%s' (including the single quotes) in place of the ID field value. The value '%s' is a placeholder for the internal ID and is assigned automatically by the Metadata Server. The ID field must be a string field 38 characters in length. In the following example, the ADMIN_TABLE has two columns: the first is "docUUID", and the second is "service_running". The column "docUUID" is the field used to relate the ADMIN_TABLE to the main internal metadata table. INSERT into <ADMIN_TABLE> (docUUID, service_running) values ('%s','No').
tablename	Name of external ADMIN_TABLE for administrator's use.

Examples for ADMIN_TABLE:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <METADATA_CONFIG>
      <WORKSPACES>
        <SDEWORKSPACE name="unique_name" server="server_name"
instance="port:5151" database="optional_database_name" user="user_name"
password="user_password" />
      </WORKSPACES>
      <ADMIN_TABLE tablename="AdminTable" idcolumn="DocUUID"
filter="Approved='Y'" insert="INSERT INTO AdminTable (DocUUID,
Approved) VALUES ('%s', 'N')">
        <RESPONSE_COLUMN columnname="Approved" />
      </ADMIN_TABLE>
      <METADATA_CONTENT validate="true" />
      <TABLE_NAME prefix="imsmetadata" />
    </METADATA_CONFIG>
  </CONFIG>
</ARCXML>
```

ADMINCMD

Used in: Application Server Administration
Parent elements: Application Server Administration

```
<ADMINCMD

    version ="1.0"  [1.0]
>

    <SERVICES... />

</ADMINCMD >
```

Bold: Attribute or child element is required.

Description:

Administers ArcIMS services from the command line.

Restrictions:

None

Notes:

- ADMINCMD is used to administer ArcIMS services from the command line. To do this, two files are used:
  - An ADMINCMD XML file
  - A batch file or script

  The ADMINCMD XML file contains the instructions for adding, starting, stopping, and removing ArcIMS services. See the examples for proper construction of an ADMINCMD XML file.

  On Windows, the batch file contains one line:

  java com.esri.aims.admincore.cmd.Exec **http://mymachine.domain.com Username Password** file **filename**

  Where:

  - **http://mymachine.domain.com** is the host machine.
  - **Username** is the user name used to log into ArcIMS Administrator or Manager.
  - **Password** is the password used to log into ArcIMS Administrator or Manager.

- **Filename** is the full pathname and name of the ADMINCMD XML file, for example, c:\arcims\axl\admincmd.xml.
- Note: the parameter "file" must be included before the filename.

The above command can also be typed on the command line in lieu of using the batch file.

On UNIX, a script file is used. An example script is:
#!/bin/csh

setenv JARHOME $AIMSHOME/Manager/lib
setenv AIMSHOST **$argv[1]**

java -cp
$JARHOME/jaxp.jar:$JARHOME/parser.jar:$JARHOME/esri_mo10.jar:
$JARHOME/esri_mo10res.jar:$JARHOME/arcims_admincore.jar:$JARHOME/a
rcims_admin.jar:
$JARHOME/jcert.jar:$JARHOME/jnet.jar:$JARHOME/jsse.jar:$JARHOME/arc
ims_resadmin.jar com.esri.aims.admincore.cmd.Exec http://$AIMSHOST admin
admin file **$argv[2]**

Where:

- **$argv[1]** is the hostname.
- **$argv[2]** is the name of the ADMINCMD XML file, for example, admincmd.xml.
- Note: the parameter "file" must be included before $argv[2].

Attribute Descriptions for ADMINCMD:

Attribute	Usage
version	Specifies version of ADMINCMD statement.

Examples for ADMINCMD:

Example 1:

```
<?xml version="1.0"?>
<ADMINCMD version="1.0">
  <SERVICES>
    <SERVICE type="stop" name="world_image" />
    <SERVICE type="remove" name="world_image" />
  </SERVICES>
</ADMINCMD>
```

ARCXML

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Query Feature Extract Geocode Metadata ArcMap
Parent elements: None

```
<ARCXML

   version ="1.1"
>

   <CONFIG... />  [Or]
   <REQUEST... />  [Or]
   <RESPONSE... />  [Or]

</ARCXML >
```

Bold: Attribute or child element is required.

Description:

The root element for all ArcXML statements. CONFIG is used in all map configuration files and viewer configuration files. REQUEST is used in all ArcIMS requests. RESPONSE is used in all ArcIMS responses.

Restrictions:

- One child element must be used in an ArcXML statement. Only one child element is permitted.

Notes:

- All ArcXML 1.1 statements are required to use a standard prolog that includes the XML version and encoding:
  <?xml version="1.0" encoding="UTF-8"?>

Attribute Descriptions for ARCXML:

Attribute	Usage
version	Specifies version of ArcXML statement.

Examples for ARCXML:

Example 1: All CONFIG, REQUEST, and RESPONSE statements begin with ARCXML.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.21" miny="18.92" maxx="-66.96"
maxy="71.41" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CITIES" visible="true" id="2">
      <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLEMARKERSYMBOL type="square" width="5" />
      </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

AVIMSWORKSPACE

Used in: CONFIG
Parent elements: WORKSPACES

```
<AVIMSWORKSPACE

    map ="string"
    name ="string"
    url ="string"
    view ="string"
>

    No Child Elements
</AVIMSWORKSPACE >
```

Bold: Attribute or child element is required.

Description:

Specifies a workspace for a map served in ArcView IMS.

Note: this element has been deprecated in ArcIMS 4.0.

Restrictions:

- Can only be used in viewer configuration files. It cannot be used in a map configuration file.

Notes:

- The values for an ArcView IMS MapName and ViewName can be found in the applet HTML file of an ArcView IMS MapCafé Web site. For example, if you named your Web site "MyPage", open "MyPageapplet.html". Look for the parameters MapName and ViewName. For more information on using ArcView IMS with ArcIMS, see ArcIMS Online.

Attribute Descriptions for AVIMSWORKSPACE:

Attribute	Usage
map	ArcView IMS MapName.
name	Workspace name. Must be unique among all data sources.
url	Server URL points to the location of the Esrimap version used with ArcView IMS. This is either the ArcIMS servlet connector (/servlet/com.esri.esrimap.Esrimap) or the ArcView IMS esrimap.dll or esrimapn.dll (/scripts/esrimap.dll, /scripts/esrimapn.dll).
view	ArcView IMS ViewName.

Examples for AVIMSWORKSPACE:

Example 1: When in a viewer configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-176.0837748424188" miny="18.9247817993164"
maxx="-52.62028121948239" maxy="83.10832214355472"
name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
      <AVIMSWORKSPACE name="av_ws-4"
url="http://mycomputer.domain.com/servlet/com.esri.esrimap.Esrimap"
view="view1" map="washoe21204240" />
      </WORKSPACES>
      <LAYER type="image"
name="mycomputer.domain.com:washoe21204240:view1" visible="true"
id="0">
      <DATASET name="mycomputer.domain.com:washoe21204240:view1"
type="image" workspace="av_ws-4" />
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

BACKGROUND

Used in: **CONFIG** **REQUEST**
Servers: Image ArcMap
Parent elements: **PROPERTIES**

```
<BACKGROUND

    color ="0,0,0 - 255,255,255"
    transcolor ="0,0,0 - 255,255,255"
>

    No Child Elements
</BACKGROUND >
```

Bold: Attribute or child element is required.

Description:

Defines color of map's background.

Restrictions:

- Valid with Image and ArcMap Servers.
- When using *transcolor*, depending on the browser, the image formats that support transparent colors vary. JPG images do not support transparent colors. The table below lists which image formats support transparent colors for different browsers.

Browser	Supported Transparent Image Formats
ArcIMS HTML Viewer in Internet Explorer	PNG8, GIF
ArcIMS HTML Viewer in Netscape 4.x	GIF
ArcIMS HTML Viewer in Netscape 6.x	PNG8, GIF
ArcExplorer 4	PNG24, PNG8, GIF
ArcIMS Java Viewers in Internet Explorer and Netscape	PNG24, PNG8, GIF

Notes:

- In order to set the background of an Image Service as transparent, *color* and *transcolor* should be set to the same value. White is 255,255,255.

Attribute Descriptions for BACKGROUND:

Attribute	Usage
color	Background color for map using RGB values.
transcolor	Transparent color in output image using RGB values. Only supported for GIF and PNG output formats.

Examples for BACKGROUND:

Example 1: Included in the PROPERTIES section for both CONFIG and REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.21" miny="18.92" maxx="-66.96"
maxy="71.41" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
        <BACKGROUND color="255,255,255" transcolor="255,255,255"/>
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CITIES" visible="true" id="2">
      <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLEMARKERSYMBOL type="square" width="5" />
      </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

BAND

Used in: **RESPONSE**
Servers: Image ArcMap
Parent elements: **BANDS**

```
<BAND

    number="integer"
    value ="integer"
>

    No Child Elements
</BAND >
```

Bold: Attribute or child element is required.

Description:

Specifies pixel value for an image band at the specified x,y coordinate.

Restrictions:

None

Notes:

None

Attribute Descriptions for BAND:

Attribute	Usage
number	Unique identifier for the band.
value	Pixel value for the band at the specified x,y coordinate.

Examples for BAND:

Example 1:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <RASTER_INFO>
      <BANDS rasterid="1">
        <BAND number="0" value="238" />
      </BANDS>
    </RASTER_INFO>
  </RESPONSE>
</ARCXML>
```

BANDS

Used in: **RESPONSE**
Servers: Image ArcMap
Parent elements: **RASTER_INFO**

<BANDS

rasterid ="*string*"
>

(m) <BAND... />

</BANDS >

(m): Child element can be used multiple times.

Description:

Parent element for listing information on one or more bands.

Restrictions:

None

Notes:

None

Attribute Descriptions for BANDS:

Attribute	Usage
rasterid	For ArcSDE raster images, the value represents the row number. When multiple files are used, such as in an image catalog, the value represents the name of the image file. For single images, the value is the name of the image file.

Examples for BANDS:

Example 1:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <RASTER_INFO>
      <BANDS rasterid="1">
        <BAND number="0" value="238" />
```

```
      </BANDS>
    </RASTER_INFO>
  </RESPONSE>
</ARCXML>
```

BUFFER

Used in: **REQUEST**
Servers: Image Query Feature ArcMap
Parent elements: **QUERY** **SPATIALFILTER** **SPATIALQUERY**

<BUFFER

When using ArcMap or Feature Server:
distance ***="double"***
bufferunits ="decimal_degrees | miles | feet | kilometers | meters" **[MAPUNITS defined in PROPERTIES]**

When using Image Server:
distance ***="double"***
bufferunits ="decimal_degrees | miles | feet | kilometers | meters" **[MAPUNITS defined in PROPERTIES]**
project ="true | false" **[true]**
>

When parent element is ***SPATIALFILTER:***
No Child Elements

When parent element is ***SPATIALQUERY:***
<SPATIALQUERY... />
<TARGETLAYER... />

</BUFFER >

Bold: Attribute or child element is required.

Description:

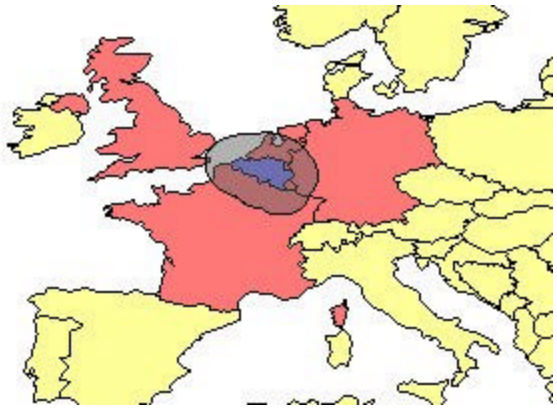
Builds a buffer region around selected features.

Restrictions:

- SPATIALQUERY used in BUFFER cannot contain another BUFFER inside.
- When using a target layer that is the same layer as the buffer layer, the resulting buffer will be different when using an ArcMap Image Service compared to Image or Feature Services. The following ArcXML shows the syntax for a buffer layer and a target layer. The buffer and target layer are the same (cities). Note that this is only an issue when the buffer layer and target layer are the same.

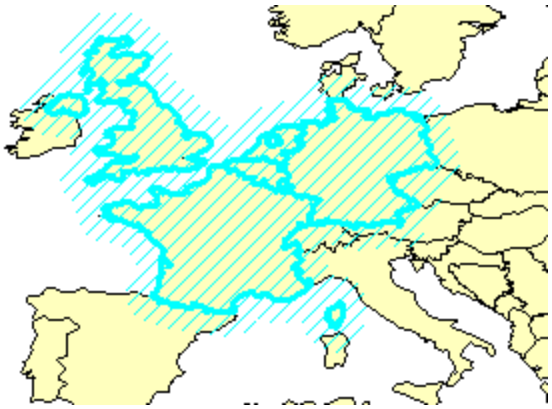
```
<!-- Set the buffer around the selected feature (Belgium) -->
<LAYER type="featureclass" name="theBuffer" visible="true"
```

```
id="buffer">
  <DATASET fromlayer="countries"/>
  <SPATIALQUERY>
    <BUFFER distance="100" bufferunits="miles" />
    <SPATIALFILTER relation="area_intersection">
      <ENVELOPE minx="4.68" miny="50.34" maxx="5.46"
maxy="50.86"/>
    </SPATIALFILTER>
  </SPATIALQUERY>
  <SIMPLERENDERER>
    <SIMPLEPOLYGONSYMBOL fillcolor="100,100,100"
filltransparency="0.5" />
  </SIMPLERENDERER>
</LAYER>

<!-- Select features in the target layer based on which countries
fall inside the buffer -->
<LAYER type="featureclass" name="theBufferTarget" visible="true"
id="buffertarget">
  <DATASET fromlayer="countries"/>
  <SPATIALQUERY>
    <BUFFER distance="100" bufferunits="miles">
      <TARGETLAYER id="countries"/>
    </BUFFER>
    <SPATIALFILTER relation="area_intersection">
      <ENVELOPE minx="4.68" miny="50.34" maxx="5.46"
maxy="50.86"/>
    </SPATIALFILTER>
  </SPATIALQUERY>
  <SIMPLERENDERER>
    <SIMPLEPOLYGONSYMBOL fillcolor="255,120,120" filltype="solid"
/>
  </SIMPLERENDERER>
</LAYER>
```



When using the **Image Server**, a feature is selected and a buffer is created around the feature. Next, the target layer is applied. All features that fall within the buffer are selected.

In this example, Belgium is selected and buffered 100 miles. The resulting buffer is around Belgium only. Since the countries layer is also the target layer, all surrounding countries that fall within the buffer are highlighted.

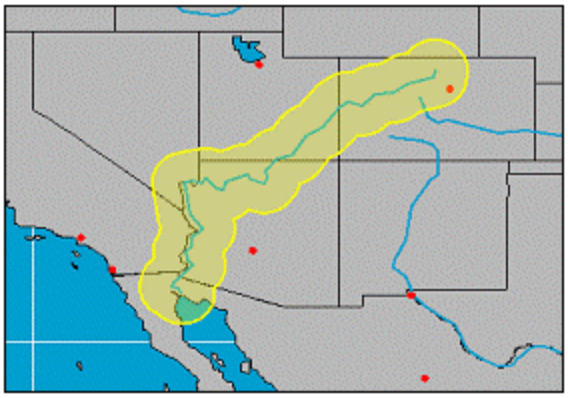


When using the **ArcMap Server**, a feature is selected and a buffer is created around the feature. Next, the target layer is applied. All features that fall within the buffer are selected. The buffer is then applied to these features as well.

In this example, Belgium is selected and buffered 100 miles. All countries within the 100-mile buffer are selected. The final 100-mile buffer is generated around all selected countries. The buffer in this example is represented by the diagonal lines.

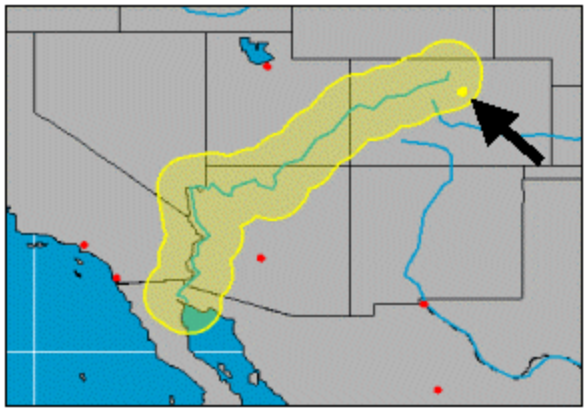
Notes:

- BUFFER can be used three ways:



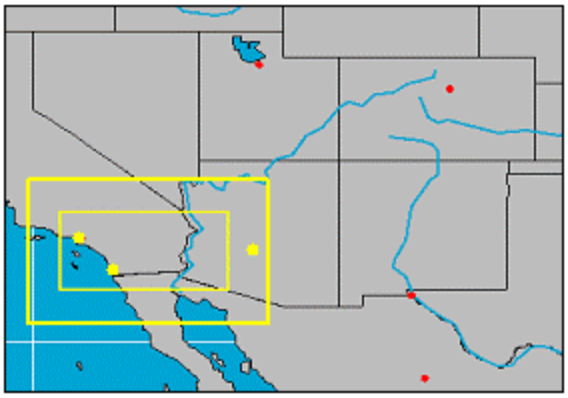
To buffer a region around one or more selected features, use BUFFER inside a SPATIALQUERY. In this example, the Colorado River is buffered.

```
<LAYER type="featureclass"
name="theRiverBuffer" id="RivBuf">
  <DATASET fromlayer="Rivers" />
  <SPATIALQUERY
where="NAME=&apos;Colorado&apos;">
    <BUFFER distance="80"
bufferunits="MILES" />
  </SPATIALQUERY>
  <SIMPLERENDERER>
    <SIMPLEPOLYGONSYMBOL
fillcolor="255,255,0"
filltransparency=".3"
boundarycolor="255,255,0"
boundarywidth="2" />
  </SIMPLERENDERER>
</LAYER>
```



To select features from the same or another layer that fall within the buffer region, use TARGETLAYER as a child element. In this example, cities within 80 miles of the Colorado River are selected. (The buffer in the picture is shown for reference. With this request, only selected cities would display. To highlight the cities and show the buffer, both the following and preceding LAYER examples must be used in a request.)

```
<LAYER type="featureclass"
name="theRiverBuffer" id="RivBuf">
  <DATASET fromlayer="Rivers" />
  <SPATIALQUERY
where="NAME=&apos;Colorado&apos;">
    <BUFFER distance="80"
bufferunits="MILES" >
      <TARGETLAYER id="Cities" />
    </BUFFER>
  </SPATIALQUERY>
  <SIMPLERENDERER>
    <SIMPLEMARKERSYMBOL
color="255,255,0" width="6" />
  </SIMPLERENDERER>
</LAYER>
```



To use a buffer as a selection tool. BUFFER is used around a defined filter within SPATIALFILTER. In this example, BUFFER is used to buffer around an envelope. The inner yellow box is the original envelope. The outer yellow box is the buffer filter. Note that these boxes would not normally display; only the selected items within the filter.

```
<LAYER type="featureclass"
name="theEvelopeBuffer"
id="EnvBuf" >
  <DATASET fromlayer="Cities" />
  <SPATIALQUERY where="POPULATION
> 1000000" >
    <SPATIALFILTER
relation="envelope_intersection">
      <ENVELOPE minx="-119"
miny="32" maxx="-113" maxy="35" />
      <BUFFER distance="100"
bufferunits="MILES" />
    </SPATIALFILTER>
  </SPATIALQUERY>
  <SIMPLERENDERER>
```

	<SIMPLEMARKERSYMBOL color="255,255,0" width="6" /> </SIMPLERENDERER> </LAYER>

- In a GET_FEATURES request, when SPATIALQUERY is a parent element to BUFFER and *subfields* is included, the field #SHAPE# or #ALL# must be included in the subfields list.
- A buffer distance of 0 can be used on polygon features. If a buffer distance of 0 is used with line or point features, no data is returned.
- Buffer results may be inaccurate on unprojected data. It is recommended to project data before applying a buffer.

Attribute Descriptions for BUFFER:

Attribute	Usage
bufferunits	Specifies units that apply to buffer.
distance	Buffer area width in buffer units.
project	Generated buffers are projected if FEATURECOORDSYS is in the map configuration file or a request. If *project* is set to false, the buffer is not projected.

Examples for BUFFER:

Example 1: When in a GET_FEATURES request.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES featurelimit="25" beginrecord="0" outputmode="xml"
geometry="FALSE" envelope="true" compact="true">
      <LAYER id="2" /> <!-- states -->
      <SPATIALQUERY subfields="#SHAPE# NAME">
        <BUFFER distance="1" >
          <TARGETLAYER id="4" /> <!-- cities -->
          <SPATIALQUERY subfields="NAME" />
        </BUFFER>
        <SPATIALFILTER relation="area_intersection" >
          <ENVELOPE minx="-16154208.3772906" miny="-4165319.9729724"
maxx="-4904885.23874079" maxy="4271672.38093997" />
        </SPATIALFILTER>
      </SPATIALQUERY>
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

Example 2: When in a GET_IMAGE request. Shows selected features and buffer. Note in this example that the DATASET fromlayer is "CITIES". This refers to the LAYER id in the map configuration file, not the LAYER name.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="867805.08306143" miny="6771243.45626185"
maxx="1576642.8482793" maxy="7302871.78017525"/>
      </PROPERTIES>
      <LAYER type="featureclass" name="theBufferTarget" visible="true"
id="buffertarget">
        <DATASET fromlayer="CITIES"/>
        <SPATIALQUERY>
          <BUFFER distance="100" bufferunits="miles">
            <TARGETLAYER id="CITIES"/>
          </BUFFER>
          <SPATIALFILTER relation="area_intersection">
            <ENVELOPE minx="867805.08306143" miny="6771243.45626185"
maxx="1576642.8482793" maxy="7302871.78017525"/>
          </SPATIALFILTER>
        </SPATIALQUERY>
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="255,0,0" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="theBuffer" visible="true"
id="buffer">
        <DATASET fromlayer="CITIES"/>
        <SPATIALQUERY>
          <BUFFER distance="100" bufferunits="miles" />
          <SPATIALFILTER relation="area_intersection">
            <ENVELOPE minx="867805.08306143" miny="6771243.45626185"
maxx="1576642.8482793" maxy="7302871.78017525"/>
          </SPATIALFILTER>
        </SPATIALQUERY>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="100,100,100" filltype="solid"
filltransparency="0.5" />
        </SIMPLERENDERER>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

CALLOUTMARKERSYMBOL

Used in: CONFIG REQUEST RESPONSE
Servers: Image
Parent elements: EXACT OTHER RANGE SIMPLELABELRENDERER

```
<CALLOUTMARKERSYMBOL

    antialiasing ="true | false"  [false]
    backcolor ="0,0,0 - 255,255,255"  [255,255,255]
    boundarycolor ="0,0,0 - 255,255,255"  [0,0,0]
    font ="Any system font"  [Arial]
    fontcolor ="0,0,0 - 255,255,255"  [0,0,0]
    fontsize ="1 - NNN"  [12]
    fontstyle ="regular | bold | italic | underline | outline | bolditalic"  [regular]
    glowing ="0,0,0 - 255,255,255"
    interval ="0 - NNN"  [10]
    outline ="0,0,0 - 255,255,255"
    shadow ="0,0,0 - 255,255,255"
    transparency ="0.0 - 1.0"  [1.0]
>

    No Child Elements
</CALLOUTMARKERSYMBOL >
```

Description:

Creates a callout box around each label.

Restrictions:

- This symbol only works with point layers.
- *Outline* and *glowing* should not be used together; use one or the other.
- Not valid with ArcMap Server.

Notes:

None

Attribute Descriptions for CALLOUTMARKERSYMBOL:

Attribute	Usage
antialiasing	Used to make edges of labels and symbols smoother. When set to "true", antialiasing is active. Note that the time to generate a map may significantly increase when antialiasing is turned on.
backcolor	Background color using RGB values.
boundarycolor	Boundary color using RGB values.
font	Font name. The name is case sensitive. If font name uses "&", use "&" instead. For example, ESRI Transportation & Civic should be written as ESRI Transportation & Civic. For Feature Services, the font must reside on the client machine or else the system default font is used.
fontcolor	Font color using RGB values.
fontsize	Font size.
fontstyle	Font style.
glowing	Glow color around text using RGB values.
interval	Distance between point and callout box; smaller number brings box closer to point.
outline	Outline color using RGB values.
shadow	Shadow color using RGB values.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.

Examples for CALLOUTMARKERSYMBOL:

Example 1: When in CONFIG or REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
         <ENVELOPE minx="-105.594842" miny="-49.955227"
maxx="75.672764" maxy="83.596039" name="Initial_Extent" />
          <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
```

```
      </WORKSPACES>
      <LAYER type="featureclass" name="CITIES" visible="true" id="2">
        <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
          <GROUPRENDERER>
            <SIMPLELABELRENDERER field="NAME">
              <CALLOUTMARKERSYMBOL font="Times New Roman"
fontstyle="italic" fontsize="24" fontcolor="0,0,255" glowing="255,0,0"
shadow="0,0,50" backcolor="0,255,0" interval="10"
boundarycolor="255,255,0" transparency="0.8" antialiasing="false" />
            </SIMPLELABELRENDERER>
            <SIMPLERENDERER>
              <SIMPLEMARKERSYMBOL color="127,27,27" type="circle"
width="16" />
            </SIMPLERENDERER>
          </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

CAPABILITIES

Used in: RESPONSE
Servers: Image Query Feature Extract Geocode ArcMap
Parent elements: ENVIRONMENT

<CAPABILITIES

forbidden **="*string*"**
>

No Child Elements
</CAPABILITIES >

Bold: Attribute or child element is required.

Description:

Used in a SERVICEINFO response to list any forbidden elements for an ArcIMS service.

Restrictions:

None

Notes:

- The list of forbidden elements is limited to the five requests: GET_FEATURES, GET_EXTRACT, GET_GEOCODE, GET_IMAGE, GET_SERVICE_INFO, GET_LAYOUT, GET_METADATA, PUBLISH_METADATA, and GET_RASTER_INFO.
- Listed elements are always delimited by a comma regardless of what is used in SEPARATORS.
- For more information on forbidden elements and user authentication of ArcIMS services, see "ArcIMS Servlet Connector" under the System Administration chapter in the ArcIMS Online Help (Help Contents).

Attribute Descriptions for CAPABILITIES:

Attribute	Usage
forbidden	List of elements that cannot be used in a request to the service. Multiple elements are always delimited by a comma.

Examples for CAPABILITIES:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <CAPABILITIES forbidden="GET_GEOCODE,GET_EXTRACT"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-71.077172" miny="42.355504" maxx="-71.034431"
maxy="42.387721" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="Streets"
id="streets">
        <FCLASS type="line">
        </FCLASS>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

CHANGE_METADATA_ACCESS

Used in: REQUEST
Servers: Metadata (Publish)
Parent elements: PUBLISH_METADATA

```
<CHANGE_METADATA_ACCESS

    docid ="string"
    private ="true | false"
>

    No Child Elements
</CHANGE_METADATA_ACCESS >
```

Bold: Attribute or child element is required.

Description:

Changes the access of a metadata document.

Restrictions:

None

Notes:

- See METADATA_ACTION for response.

Attribute Descriptions for CHANGE_METADATA_ACCESS:

Attribute	Usage
docid	String used to uniquely identify a document. The client used to publish the metadata is responsible for creating the document ID. This ID is automatically assigned when using ArcCatalog. If another client is used, GET_UUID can be used to request a valid ID. The format for an ID is the following: {HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH} where H is a hexadecimal digit (0-9,a-f,A-F). The ID is limited to 38 characters.
private	Determines whether document is viewable for all users. Use "false" if the document is viewable by all users. Use "true" when the document is viewable only by the document owner.

Examples for CHANGE_METADATA_ACCESS:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <PUBLISH_METADATA>
      <CHANGE_METADATA_ACCESS docid="{C64D8F38-82B4-11D5-99C2-
000086460FA0}" private="true" />
    </PUBLISH_METADATA>
  </REQUEST>
</ARCXML>
```

CHARTSYMBOL

Used in: CONFIG REQUEST RESPONSE
Servers: Image Feature
Parent elements: EXACT OTHER RANGE SIMPLELABELRENDERER

```
<CHARTSYMBOL

    antialiasing ="true | false"  [false]
    maxsize ="1 - NNN"
    maxvalue ="1 - NNN"
    minsize ="1 - NNN"
    minvalue ="1 - NNN"
    mode ="pie | bar"  [pie]
    outline ="0,0,0 - 255,255,255"  [none]
    shadow ="0,0,0 - 255,255,255"
    size ="1 - NNN"
    sizefield ="string"
    transparency ="0.0 - 1.0"  [1.0]
>

    (m) <CHARTVALUE... />

</CHARTSYMBOL >
```

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Symbolizes features with bar or pie diagrams.

Although chart symbols draw on the map, no information about the chart symbols is displayed in the legend. ArcIMS 4 does not support displaying chart values in the legend.

Restrictions:

- The size of a chart can be determined in one of three ways:
    1. Using the *size* attribute.
    2. Using the *sizefield* attribute. This method works only with Image Services. It does not work with Feature Services.
    3. Using the *minsize*, *maxsize*, *minvalue*, and *maxvalue* attributes.

  One of the three options must be used or no charts are drawn. *Size* ranks first in priority, and it takes precedence over the other attributes even if they are included. *Sizefield* is second in priority, and it takes precedence over the other attributes if

size is not present. *Minsize*, *maxsize*, *minvalue*, and *maxvalue* rank third in priority and are acknowledged only when *size* and *sizefield* are not present.

- Not valid with ArcMap Server.

Notes:

- A field listed in CHARTSYMBOL *sizefield* must also be listed in SIMPLELABELRENDERER *field* or VALUEMAPLABELRENDERER *labelfield*.
- The attributes *minsize* and *maxsize* determine the minimum and maximum size of a chart, respectively. For each feature, the size of the chart is relative to other features based on the minimum and maximum values. These values are set using *minvalue* and *maxvalue*.

  If a feature has a value less than or equal to *minvalue*, then the chart is drawn at *minsize*. If a feature has a value greater than or equal to *maxvalue*, then the chart is drawn at *maxsize*. In the following example, the values are based on population. If a feature has a population less than or equal to 10,000, then the size of the chart is 10 pixels. If the feature has a population greater than or equal to 200,000, then the size of the chart is 50 pixels.

  <CHARTSYMBOL minsize="10" minvalue="10000" maxsize="50" maxvalue="200000" >

  All other features are assigned a chart size between 10 and 50 pixels based on population. A feature with a population of 100,000 will have a chart approximately 30 pixels in size. A feature with a population of 1,000,000 is assigned a chart size of 50 pixels since that is the maximum size allowed using *maxsize*.

- If the size assigned to a chart is very large, the chart may not display. If you find that no charts are displaying, try a smaller chart size.
- This element is not available from the ArcIMS Author interface.

Attribute Descriptions for CHARTSYMBOL:

Attribute	Usage
antialiasing	Used to make edges of labels and symbols smoother. When set to "true", antialiasing is active. Note that the time to generate a map may significantly increase when antialiasing is turned on.
maxsize	Maximum size of chart if *size* or *sizefield* is not used.
maxvalue	Maximum value that corresponds to the maximum chart size in *maxsize*.
minsize	Minimum size of chart if *size* or *sizefield* is not used.
minvalue	Minimum value that corresponds to the minimum chart size in *minsize*.

mode	Type of chart: pie or bar.
outline	Outline color of charts using RGB values.
shadow	Shadow color using RGB values.
size	Size of charts. All charts are the same size.
sizefield	The field in the database containing the size of the chart. The field can be in the layer table or in a joined table. • For shapefiles with no joined tables, the field can be referenced using the short format. sizefield="AREA" • For shapefiles with joined tables, the name of the joined table must be included along with the field. sizefield="JOINEDTABLE.AREA" Files with joined fields cannot be read locally in ArcIMS Author or ArcExplorer 4. • For ArcSDE layers without joined tables, the field can be referenced using the short format. sizefield="AREA" The full long name can also be used, but when a long name is used, the file cannot be read locally in ArcIMS Author or ArcExplorer 4. sizefield="ARCSDENAME.TABLE.AREA" • For ArcSDE layers with joined tables, joined fields must be referenced using the full long format. sizefield="ARCSDENAME.TABLE.AREA" Files with joined fields cannot be read locally in ArcIMS Author or ArcExplorer 4. Works only with Image Services. Does not work with Feature Services.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.

Examples for CHARTSYMBOL:

Example 1: Using minsize, minvalue, maxsize, and maxvalue attributes.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
    <PROPERTIES>
```

```
      <ENVELOPE minx="-127.87777503188939" miny="30.378245451392196"
maxx="-101.1629831289576" maxy="48.55733555119212"
name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
    </PROPERTIES>
    <WORKSPACES>
      <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>" />
    </WORKSPACES>
    <LAYER type="featureclass" name="States" visible="true" id="0">
      <DATASET name="states" type="polygon" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL fillcolor="127,127,27" />
          </SIMPLERENDERER>
          <SIMPLELABELRENDERER field="POP1999 POP1990">
            <CHARTSYMBOL minsize="10" minvalue="1000000" maxsize="50"
maxvalue="7000000" outline="255,255,255" shadow="0,0,0"
transparency="1.0" >
              <CHARTVALUE lookupfield="POP1990" color="255,0,0" />
              <CHARTVALUE lookupfield="POP1999" color="0,0,255" />
            </CHARTSYMBOL>
          </SIMPLELABELRENDERER>
        </GROUPRENDERER>
      </LAYER
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: Using an ArcSDE layer and the size attribute.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-127.34083168343714" miny="27.246314948863663"
maxx="-97.31362835016226" maxy="50.018623517052" name="Initial_Extent"
/>
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SDEWORKSPACE name="sde_ws-4" server="ARCSDENAME"
NAME="port:5151" user="SDE" encrypted="true" password="OCOBLVWKFCAEHC"
/>
      </WORKSPACES>
      <LAYER type="featureclass" name="SDE.US_STATES" visible="true"
id="0">
      <DATASET name="SDE.US_STATES" type="polygon" workspace="sde_ws-4"
/>
```

```
      <GROUPRENDERER>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="27,27,227" />
        </SIMPLERENDERER>
          <SIMPLELABELRENDERER field="SDE.US_STATES.POP1999
SDE.US_STATES.POP1990">
            <CHARTSYMBOL size="30" >
              <CHARTVALUE lookupfield="SDE.US_STATES.POP1990"
color="255,0,0" />
              <CHARTVALUE lookupfield="SDE.US_STATES.POP1999"
color="0,0,255" />
            </CHARTSYMBOL>
          </SIMPLELABELRENDERER>
      </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 3: Using sizefield with the field in a joined DBF file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
    <PROPERTIES>
      <ENVELOPE minx="-127.87777503188939" miny="30.378245451392196"
maxx="-101.1629831289576" maxy="48.55733555119212"
name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
    </PROPERTIES>
    <WORKSPACES>
      <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>" />
    </WORKSPACES>
    <LAYER type="featureclass" name="States" visible="true" id="0">
      <DATASET name="states" type="polygon" workspace="shp_ws-0" />
        <SPATIALQUERY where="states.SUB_REGION='Mtn'"
joinexpression="To=[states.STATE_FIPS],From=[relatefile.STATE_FIPS],Typ
e=[scan]" jointables="relatefile" />
       <GROUPRENDERER>
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL fillcolor="127,127,27" />
          </SIMPLERENDERER>
          <SIMPLELABELRENDERER  field="POP1990 POP1999
RELATEFILE.CHARTSIZE">
            <CHARTSYMBOL mode="bar" sizefield="RELATEFILE.CHARTSIZE"
transparency="1.0" >
              <CHARTVALUE lookupfield="POP1990" color="255,0,0" />
              <CHARTVALUE lookupfield="POP1999" color="0,0,255" />
            </CHARTSYMBOL>
```

```
          </SIMPLELABELRENDERER>
        </GROUPRENDERER>
      </LAYER
    </MAP>
  </CONFIG>
</ARCXML>
```

CHARTVALUE

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Feature
Parent elements: **CHARTSYMBOL**

<CHARTVALUE

lookupfield ="***string***"
color ="0,0,0 - 255,255,255" **[0,0,0]**
lower ="*integer*"
upper ="*integer*"
value ="*integer*"
>

No Child Elements
</CHARTVALUE >

Bold: Attribute or child element is required.

Description:

Sets which fields are used in pie and bar diagrams.

Although chart symbols draw on the map, no information about the chart symbols is displayed in the legend. ArcIMS 4 does not support displaying chart values in the legend.

Restrictions:

- The attribute *lookupfield* is not required when *value* is used. It is required in all other cases.
- In a Feature Service, *lookupfields* cannot include fields from a joined table. In an Image Service, fields from joined tables can be used.
- Not valid with ArcMap Server.

Notes:

- All fields listed in CHARTVALUE *lookupfield* must also be listed in SIMPLELABELRENDERER *field* or VALUEMAPLABELRENDERER *labelfield*.
- The attributes *lower* and *upper* are used together to change the color of a chart segment depending on the value for a feature. For example, a segment might be blue for population less than 1,000,000 and red for population greater than 1,000,000. For each range, a new CHARTSYMBOL should be used for referencing the same *lookupfield* such as in the following example:

```
<SIMPLELABELRENDERER field="POP1999" >
  <CHARTSYMBOL size="30" transparency="1.0" >
    <CHARTVALUE lookupfield="POP1999" color="0,0,255" lower="0"
upper="1000000" />
    <CHARTVALUE lookupfield="POP1999" color="255,0,0"
lower="1000001" upper="100000000" />
  </CHARTSYMBOL>
</SIMPLELABELRENDERER>
```

- The attribute *value* is used to change the color of a chart segment depending on user-assigned integer values. All values within the CHARTSYMBOL group are a ratio of the sum of the values. For example, assume two CHARTVALUEs are used to build a chart, and the values are 14018000 and 15743000. The sum of these values is 29761000. Therefore, the size of the first chart segment is 14018000 / 29761000, or 47 percent of the chart. The second segment is 53 percent.

  In the following example, values were calculated ahead of time for populations under 30 and 30 and over for California. These two values were then assigned to two CHARTVALUEs. The VALUEMAPLABELRENDERER *labelfield* must be present even though it is not used. For the ArcXML string to parse properly, a valid field name in *labelfield* must be included. If SIMPLELABELRENDERER were used, *field* would be used instead. Again, a valid field name must be included. In CHARTVALUE, no *lookupfield* is needed.

```
<VALUEMAPLABELRENDERER lookupfield="STATE_NAME"
labelfield="POP1999" >
  <EXACT value="California">
    <CHARTSYMBOL size="50" >
      <CHARTVALUE color="255,0,0" value="14018000" />
      <CHARTVALUE color="0,0,255" value="15743000" />
    </CHARTSYMBOL>
  </EXACT>
</VALUEMAPLABELRENDERER>
```

- The *lookupfield* order must be the same order as fields listed in SIMPLELABELRENDERER *field* or VALUEMAPLABELRENDERER *labelfield*.
- This element is not available from the ArcIMS Author interface.

Attribute Descriptions for CHARTVALUE:

Attribute	Usage
color	Color used for segment of pie or bar chart using RGB values.
lookupfield	Lookup field for chart segment. The field can be in the layer table or a joined table. • For shapefiles with no joined tables, the field can be referenced using the short format.

	lookupfield="AREA" • For shapefiles with joined tables, the name of the joined table must be included along with the field. lookupfield="JOINEDTABLE.AREA" Files with joined fields cannot be read locally in ArcIMS Author or ArcExplorer 4. • For ArcSDE layers without joined tables, the field can be referenced using the short format. lookupfield="AREA" The full long name can also be used, but when a long name is used, the file cannot be read locally in ArcIMS Author or ArcExplorer 4. lookupfield="ARCSDENAME.TABLE.AREA" • For ArcSDE layers with joined tables, joined fields must be referenced using the full long format. lookupfield="ARCSDENAME.TABLE.AREA" Files with joined fields cannot be read locally in ArcIMS Author or ArcExplorer 4. Joined tables are valid only with Image Services. They do not work with Feature Services.
lower	Used together with *upper* to determine the minimum value for drawing a chart segment.
upper	Used together with *lower* to determine the maximum value for drawing a chart segment.
value	Used to change the color of a chart segment depending on user-assigned integer values. All values within the CHARTSYMBOL group are a ratio of the sum of the values.

Examples for CHARTVALUE:

Example 1: Using an ArcSDE layer.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-127.34083168343714" miny="27.246314948863663"
maxx="-97.31362835016226" maxy="50.018623517052" name="Initial_Extent"
/>
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
```

```
      <WORKSPACES>
        <SDEWORKSPACE name="sde_ws-4" server="ARCSDENAME"
instance="port:5151" user="SDE" encrypted="true"
password="OCOBLVWKFCAEHC" />
      </WORKSPACES>
      <LAYER type="featureclass" name="SDE.US_STATES" visible="true"
id="0">
      <DATASET name="SDE.US_STATES" type="polygon" workspace="sde_ws-4"
/>
      <GROUPRENDERER>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="27,27,227" />
        </SIMPLERENDERER>
          <SIMPLELABELRENDERER field="SDE.US_STATES.POP1999
SDE.US_STATES.POP1990">
            <CHARTSYMBOL size="30" >
              <CHARTVALUE lookupfield="SDE.US_STATES.POP1990"
color="255,0,0" />
              <CHARTVALUE lookupfield="SDE.US_STATES.POP1999"
color="0,0,255" />
            </CHARTSYMBOL>
          </SIMPLELABELRENDERER>
      </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: Using upper and lower attributes.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
    <PROPERTIES>
      <ENVELOPE minx="-127.87777503188939" miny="30.378245451392196"
maxx="-101.1629831289576" maxy="48.55733555119212"
name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
    </PROPERTIES>
    <WORKSPACES>
      <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>" />
    </WORKSPACES>
    <LAYER type="featureclass" name="States" visible="true" id="0">
      <DATASET name="states" type="polygon" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL fillcolor="127,127,27" />
          </SIMPLERENDERER>
```

```
          <SIMPLELABELRENDERER field="POP1990 POP1999">
            <CHARTSYMBOL size="40" transparency="1.0" >
              <CHARTVALUE lookupfield="POP1990" color="255,0,0"
lower="0" upper="1000000" />
              <CHARTVALUE lookupfield="POP1990" color="155,0,0"
lower="1000001" upper="7000000" />
              <CHARTVALUE lookupfield="POP1990" color="55,0,0"
lower="7000001" upper="200000000" />
              <CHARTVALUE lookupfield="POP1999" color="0,0,255" />
            </CHARTSYMBOL>
          </SIMPLELABELRENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 3: Using the value attribute.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178" miny="18" maxx="-66" maxy="71"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="Path to USA
ESRIDATA" />
      </WORKSPACES>
      <LAYER type="featureclass" name="STATES" visible="true" id="0">
        <DATASET name="STATES" type="polygon" workspace="shp_ws-0" />
        <GROUPRENDERER>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="27,127,227"/>
        </SIMPLERENDERER>
        <VALUEMAPLABELRENDERER lookupfield="STATE_NAME"
labelfield="POP1999" >
          <EXACT value="Nevada">
              <CHARTSYMBOL size="30" >
                <CHARTVALUE  color="255,0,0" value="526000" />
                <CHARTVALUE  color="0,0,255" value="676000" />
              </CHARTSYMBOL>
          </EXACT>
          <EXACT value="California">
              <CHARTSYMBOL size="50" >
                <CHARTVALUE lookupfield="MALES"  color="255,0,0"
value="14018000" />
```

```
                <CHARTVALUE lookupfield="MALES" color="0,0,255"
value="15743000" />
              </CHARTSYMBOL>
            </EXACT>
          </VALUEMAPLABELRENDERER>
          </GROUPRENDERER>
        </LAYER>
      </MAP>
    </CONFIG>
</ARCXML>
```

CLEANUP

Used in: Application Server RESPONSE
Parent elements: SERVICE

<CLEANUP

interval ="*integer*"
>

No Child Elements
</CLEANUP >

Bold: Attribute or child element is required.

Description:

Interval for ArcIMS Tasker to delete map images in the Output directory.

Restrictions:

None

Notes:

None

Attribute Descriptions for CLEANUP:

Attribute	Usage
interval	Interval for the number of minutes between the deletion of map image files from the Output directory.

Examples for CLEANUP:

Example 1:

```
<?xml version="1.0"?>
<ARCXML version="1.1">
<RESPONSE>
  <SERVICES>
    <SERVICE name="world_image" servicegroup="ImageServer1"
access="PUBLIC" type="ImageServer" desc=""  status="ENABLED" >
      <ENVIRONMENT>
        <LOCALE country="US" language="en" />
        <UIFONT name="Arial" />
      </ENVIRONMENT>
      <CLEANUP interval="20" />
```

```
    </SERVICE>
  </SERVICES>
</RESPONSE>
</ARCXML>
```

COLLECTION_INFO

Used in: RESPONSE
Servers: Metadata (Browse)
Parent elements: METADATA

<COLLECTION_INFO >
No Attributes

(m) **<TAG... />** *[Or]*
(m) **<WORD... />** *[Or]*

</COLLECTION_INFO >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Returns information describing a collection of documents in the metadata repository.

Restrictions:

- The response returns either a collection of WORD elements or TAG elements, but not both.

Notes:

- See GET_COLLECTION_INFO for request.

Examples for COLLECTION_INFO:

Example 1: When collection="tags" in a GET_COLLECTION_INFO request.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <METADATA>
      <COLLECTION_INFO>
        <TAG value="metadata/Binary/Thumbnail/Data" />
        <TAG value="metadata/dataqual/attracc/attraccr" />
        <TAG value="metadata/dataqual/complete" />
         ...
      </COLLECTION_INFO>
    </METADATA>
  </RESPONSE>
</ARCXML>
```

Example 2: When collection="words" in a GET_COLLECTION_INFO request.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <METADATA>
      <COLLECTION_INFO>
        <WORD value="data" />
        <WORD value="esri" />
        <WORD value="geographic" />
         ...
      </COLLECTION_INFO>
    </METADATA>
  </RESPONSE>
</ARCXML>
```

CONFIG

Used in: CONFIG
Parent elements: ARCXML

<CONFIG >
No Attributes

*When parent element is **CONFIG in a map configuration file:***
<ENVIRONMENT... />
<MAP... />

*When parent element is **CONFIG in a metadata configuration file:***
<METADATA_CONFIG... />

*When parent element is **CONFIG in a viewer configuration file:***
<ENVIRONMENT... />
<MAP... />
<OVERVIEWMAP... />
<SCALEBAR... />

</CONFIG >

Bold: Attribute or child element is required.

Description:

The main element for defining a configuration file.

Restrictions:

None

Notes:

- For more information on map and viewer configuration files, see Using Configuration Files.

Examples for CONFIG:

Example 1: When in a map configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
```

```
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.21" miny="18.92" maxx="-66.96"
maxy="71.41" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CITIES" visible="true" id="2">
      <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLEMARKERSYMBOL type="square" width="5" />
      </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When in a viewer configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-178.215027" miny="18.924782" maxx="-66.969849"
maxy="71.406647" name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <FEATURESERVERWORKSPACE name="ifs_ws-0"
url="http://myserver.domain.com/servlet/com.esri.esrimap.Esrimap"
service="world" />
      </WORKSPACES>
      <LAYER type="featureclass" name="STATES" visible="true" id="0">
      <DATASET name="0" type="polygon" workspace="ifs_ws-0" />
      </LAYER>
      <LAYER type="featureclass" name="ROADS" visible="true" id="1">
      <DATASET name="3" type="line" workspace="ifs_ws-0" />
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 3: When in the default.axl file used in the ArcIMS Java Viewers.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-180.0" miny="-90.0" maxx="185.6901927947995"
maxy="90.0" name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
      <IMAGESERVERWORKSPACE name="mapper_ws-0"
url="http://mymachine.domain.com/servlet/com.esri.esrimap.Esrimap"
service="background" />
      <FEATURESERVERWORKSPACE name="ifs_ws-1"
url="http://mymachine.domain.com/servlet/com.esri.esrimap.Esrimap"
service="world" />
      </WORKSPACES>
      <LAYER type="image" name="background" visible="true" id="0">
      <DATASET name="background" type="image" workspace="mapper_ws-0"
/>
      </LAYER>
      <LAYER type="featureclass" name="Countries" visible="true"
id="1">
      <DATASET name="1" type="polygon" workspace="ifs_ws-1" />
      </LAYER>
    </MAP>
    <SCALEBAR backcolor="192,192,192" fontcolor="0,0,0"
mapunits="decimal_degrees" scaleunits="feet" screenunits="inches" />
  </CONFIG>
</ARCXML>
```

CONTENT_INFO

Used in: RESPONSE
Servers: Metadata (Browse)
Parent elements: METADATA

<CONTENT_INFO

validate ="true | false" [true]
>

No Child Elements
</CONTENT_INFO >

Bold: Attribute or child element is required.

Description:

Response for whether content is validated by clients.

Restrictions:

None

Notes:

- See GET_CONTENT_INFO for request.
- The following items of content are central to the operations of searching and viewing search results in the Metadata Explorer. It is recommended that all published documents contain these items: Title, Publisher, Content type, Data theme, and Extent. These items are required if *validate* is "true" in METADATA_CONTENT.

Attribute Descriptions for CONTENT_INFO:

Attribute	Usage
validate	Validation is set to "true" by default. If set to "false", clients do not validate which items of content are available.

Examples for CONTENT_INFO:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <METADATA>
      <CONTENT_INFO validate="true" />
    </METADATA>
  </RESPONSE>
</ARCXML>
```

COORDS

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Query Feature Extract ArcMap
Parent elements: **HOLE** **MULTIPOINT** **PATH** **RING**

<COORDS >
No Attributes

No Child Elements
</COORDS >

Description:

Provides a compact way to represent points for features.

Restrictions:

None

Notes:

- The separators between the x,y coordinates and the coordinate pairs can be changed. The default separator between two coordinates is a space, and the default separator between coordinate pairs is a semicolon (;). For more information see SEPARATORS.
- COORDS uses a compact method for listing the x,y coordinate pairs of a feature in the format of

```
<RING>
   <COORDS> x1 x2;y1 y2;...xn yn </COORDS>
</RING>
```

  The equivalent noncompact method of listing x,y coordinate pairs is to use the POINT element as follows:

```
<RING>
  <POINT x="x1" y="y1" />
  <POINT x="x2" y="y2" />
  <POINT x="xn" y="yn" />
</RING>
```

- For the different options of representing geometry in an acetate layer, see OBJECT.

Examples for COORDS:

Example 1: Used to define a spatial filter in CONFIG or REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-152.9" maxx="180.0" maxy="153.0"
/>
      <LEGEND title="Legend" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="Featureclass" name="Cities" visible="true" id="0">
        <DATASET name="cities" type="point" workspace="shp_ws-0" />
        <SPATIALQUERY>
          <SPATIALFILTER relation="area_intersection">
            <POLYGON>
              <RING>
                <COORDS>-133.15 78.07;-131.09 74.70;-128.10 76.38;-
128.10 76.38;-133.15 78.07</COORDS>
              </RING>
            </POLYGON>
        </SPATIALFILTER>
        </SPATIALQUERY>
        <SIMPLERENDERER>
          <TRUETYPEMARKERSYMBOL transparency="0.5" glowing="0,255,255"
shadow="0,0,0" font="ESRI Cartography" character="252"
fontcolor="255,255,0" fontsize="16" angle="90" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When using COORDS in an acetate layer.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <IMAGESIZE width="800" height="600" />
      </PROPERTIES>
```

```
      <LAYER type="acetate" name="Selectedmark" id="acetate">
        <OBJECT units="database">
          <SIMPLELINESYMBOL color="0,0,0" width="6" />
          <POLYLINE>
            <PATH>
              <COORDS>-2.1079549837513
19.99815142335011;28.99468788980437 15.88488432940315;55.99468788980437
35.88488432940315 </COORDS>
            </PATH>
          </POLYLINE>
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 3: When in a FEATURES response.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
<RESPONSE>
  <FEATURES>
    <FEATURE>
      <ENVELOPE minx="-89.04" miny="15.91" maxx="-87.90"
maxy="18.51"/>
      <FIELDS AREA="8543.292" NAME="Belize" ABBREVNAME="Belize"
FIPS_CODE="BH"   SHAPE ="[Geometry]" ID ="105" />
      <POLYGON>
        <RING>
          <COORDS>-88.11 18.51;-88.19 18.32;-87.92 18.40;-87.90 ...-
88.11 18.51</COORDS>
        </RING>
      </POLYGON>
    </FEATURE>
    <FEATURECOUNT count="1" hasmore="false" />
  </FEATURES>
</RESPONSE>
</ARCXML>
```

COORDSYS

Used in: **CONFIG** **REQUEST**
Servers: Image Feature Extract ArcMap
Parent elements: **GET_RASTER_INFO** **LAYER** **OBJECT**

<COORDSYS

When using ArcMap Server:
id **="*integer*"**
string **="*string*"**

When using Image, Extract, or Feature Server:
id **="*integer*"**
string **="*string*"**
datumtransformid ="*integer*"
datumtransformstring ="*string*"
>

No Child Elements
</COORDSYS >

Bold: Attribute or child element is required.

Description:

Defines the projection coordinate system of a layer. COORDSYS cannot be used to project a layer; its purpose is to provide the metadata for the layer. FEATURECOORDSYS and FILTERCOORDSYS are used to project the layers to a specified projection.

Restrictions:

- Must use either *id* or *string*, but not both.
- For datum transformations either *datumtransformid* or *datumtransformstring* is used, but not both.
- In ArcMap Image Services, *datumtransformid* or *datumtransformstring* are not valid.

Notes:

- For a complete list of supported IDs and definition strings, see:
  - Projected Coordinate Systems Listing [Sorted by projection ID] [Sorted by name]
  - Geographic Coordinate Systems Listing [Sorted by projection ID] [Sorted by name]
  - Datum Transformation Listing [Sorted by projection ID] [Sorted by name]

- When using definition strings, the quotes in the string must be changed to " so the ArcIMS Spatial Server can interpret the string correctly. For example, the definition string for World Mollweide is:

  PROJCS["World_Mollweide",GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",0.0174532925199433295]],PROJECTION["Mollweide"],PARAMETER["False_Easting",0],PARAMETER["False_Northing",0],PARAMETER["Central_Meridian",0],UNIT["Meter",1]]

  Once the quotes have been changed, the string looks like this:

  PROJCS["World_Mollweide",GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",0.0174532925199433295]],PROJECTION["Mollweide"],PARAMETER["False_Easting",0],PARAMETER["False_Northing",0],PARAMETER["Central_Meridian",0],UNIT["Meter",1]]

- The attributes *datumtransformid* and *datumtransformstring* are used when datum transformation information needs to be included. Only datum transformations to and from WGS 1984 are supported.
  - When these attributes are used with COORDSYS and FILTERCOORDSYS, the datum transformation is from a non-WGS 1984 datum to WGS 1984. For example, Pulkovo_1942_To_WGS_1984 (datumtransformid="8157") transforms data from Pulkovo 1942 to WGS 1984.
  - When these attributes are used with FEATURECOORDSYS, the datum transformation is from WGS 1984 to a non-WGS 1984 datum. In the above example, the datum transformation is from WGS 1984 to Pulkovo 1942.

- When using COORDSYS in an acetate layer, the COORDSYS element should be placed inside the OBJECT element rather than the LAYER element.
- COORDSYS can be used to designate the current projection of an image. Image layers can be projected along with feature and acetate layers.
- COORDSYS is always required in order to project image or acetate layers.
- COORDSYS is not needed if no datum transformation information is needed for the layer and:
  - A *.prj file is available for a shapefile.
  - A *.prj file is available for a coverage in ArcSDE for Coverages.
  - A spatial references table is used in ArcSDE.

  Datum transformation information is not included in *.prj files and spatial reference tables. Therefore, COORDSYS must be included if a datum transformation is needed.

- If a layer does not project, double check that a *.prj file or spatial reference table exists for the layer. If not, COORDSYS must be included.
- For more information on the projection elements, see Using Projection Elements.

Attribute Descriptions for COORDSYS:

Attribute	Usage
datumtransformid	Datum transformation ID. Use either *datumtransformid* or *datumstransformstring*, but not both.
datumtransformstring	Datum transformation definition string. Use either *datumtransformid* or *datumstransformstring*, but not both.
id	Projected or geographic coordinate system ID. Use either *id* or *string*, but not both.
string	Projected or geographic coordinate system definition string. Use either *id* or *string*, but not both.

Examples for COORDSYS:

Example 1: Using a coordinate system ID.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <MAPUNITS units="meters" />
        <FEATURECOORDSYS id="54008" />
        <FILTERCOORDSYS id="4326"  />
      </PROPERTIES>
      <WORKSPACES>
      <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="Cntry94" visible="true" id="0">
        <DATASET name="Cntry94" type="polygon" workspace="shp_ws-0" />
        <COORDSYS id="4326" />
        <SIMPLERENDERER>
           <SIMPLEPOLYGONSYMBOL filltransparency="1.0"
fillcolor="27,127,127" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: Using a coordinate system definition string.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-15066114" miny="-9182334" maxx="15311928"
maxy="8551044" />
        <MAPUNITS units="meters" />
        <FEATURECOORDSYS id="54008" />
        <FILTERCOORDSYS id="54008"/>
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to data>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="Cntry94_Mollweide"
visible="true" id="0">
      <DATASET name="Cntry94_Mollweide" type="polygon"
workspace="shp_ws-0" />
      <COORDSYS
string="PROJCS[&quot;World_Mollweide&quot;,GEOGCS[&quot;GCS_WGS_1984&qu
ot;,DATUM[&quot;D_WGS_1984&quot;,SPHEROID[&quot;WGS_1984&quot;,6378137,
298.257223563]],PRIMEM[&quot;Greenwich&quot;,0],UNIT[&quot;Degree&quot;
,0.0174532925199433295]],PROJECTION[&quot;Mollweide&quot;],PARAMETER[&qu
ot;False_Easting&quot;,0],PARAMETER[&quot;False_Northing&quot;,0],PARAM
ETER[&quot;Central_Meridian&quot;,0],UNIT[&quot;Meter&quot;,1]]" />
      <SIMPLERENDERER>
         <SIMPLEPOLYGONSYMBOL filltransparency="1.0"
fillcolor="27,127,127" />
      </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 3: Using a datum transformation from Pulkovo 1942 to WGS 1984 (datumstransformid="8206"). The overall MapService has a FEATURECOORDSYS in geographic coordinates (id="4326").

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
```

```
        <ENVELOPE minx="25.2483" miny="45.6663"
maxx="25.49986079043528" maxy="45.833288705864874"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
        <FEATURECOORDSYS id="4326" />
        <FILTERCOORDSYS id="4326" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-2" directory="<path to data>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="w84shp" visible="true" id="0">
        <DATASET name="w84shp" type="point" workspace="shp_ws-2" />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="0,227,27" width="10" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="pulk42" visible="true" id="1">
        <DATASET name="pulk42" type="point" workspace="shp_ws-2" />
        <COORDSYS id="4284" datumtransformid="8206" />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="0,0,255" width="6" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

DATAFRAME

Used in: REQUEST
Servers: ArcMap
Parent elements: GET_LAYOUT

```
<DATAFRAME

    id ="string"
>

    <ENVELOPE... />
    <FEATURECOORDSYS... />
    <FILTERCOORDSYS... />
    <SCALE... />

</DATAFRAME>
```

Description:

Provides framework for listing properties of a specified data frame in an ArcMap layout.

Restrictions:

- Either ENVELOPE or SCALE may be used, but not both.
- If multiple dataframes in the *.mxd file have the same name, only the first dataframe with the name is drawn.
- Valid with ArcMap Server only.

Notes:

None

Attribute Descriptions for DATAFRAME:

Attribute	Usage
id	Name of the data frame. Data frame names are listed in the SERVICEINFO response under DATAFRAMEINFO *name*.

Examples for DATAFRAME:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_LAYOUT>
      <PROPERTIES>
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
        <FILTERCOORDSYS id="54030" />
        <FEATURECOORDSYS id="54030" />
        <IMAGESIZE width="800" height="600" />
        <OUTPUT type="JPG" />
      </PROPERTIES>
      <DATAFRAME id="Layers" >
        <FILTERCOORDSYS id="4326" />
        <FEATURECOORDSYS id="4326" />
        <ENVELOPE minx="-121" miny="36" maxx="-112" maxy="44" />
      </DATAFRAME>
    </GET_LAYOUT>
  </REQUEST>
</ARCXML>
```

DATAFRAMEINFO

Used in: **RESPONSE**
Servers: ArcMap
Parent elements: **SERVICEINFO**

<DATAFRAMEINFO >
No Attributes

(m) <LAYERINFO... />
<PROPERTIES... />

</DATAFRAMEINFO >

(m): Child element can be used multiple times.

Description:

Lists information for dataframes in an ArcMap layout.

Restrictions:

None

Notes:

None

Examples for DATAFRAMEINFO:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US"/>
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular"/>
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576"/>
        <CAPABILITIES forbidden=""/>
      </ENVIRONMENT>
      <LAYOUTINFO pageunits="inches">
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
      </LAYOUTINFO>
      <DATAFRAMEINFO name="Layers">
        <PROPERTIES>
          <FEATURECOORDSYS id="4269"/>
          <FILTERCOORDSYS id="4269"/>
```

```
          <MAPUNITS units="decimal_degrees"/>
          <BACKGROUND color="255,255,255"/>
          <ENVELOPE minx="-127.714285386824" miny="-6.58527935381"
maxx="-63.9877554315853" maxy="81.1449102179015" name="Initial_Extent"
/>
        </PROPERTIES>
        <LAYERINFO type="featureclass" name="states" id="2"
visible="true">
          <FCLASS type="polygon"></FCLASS>
        </LAYERINFO>
        <LAYERINFO type="featureclass" name="rivers" id="1"
visible="true">
          <FCLASS type="line"></FCLASS>
        </LAYERINFO>
        <LAYERINFO type="featureclass" name="cities" id="0"
visible="true">
          <FCLASS type="point"></FCLASS>
        </LAYERINFO>
      </DATAFRAMEINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

DATASET

Used in: **CONFIG** **REQUEST**
Servers: Image Query Feature Extract ArcMap
Parent elements: **LAYER**

<DATASET

When parent element is ***LAYER and the layer type is featureclass:***
name **="*string*"**
type **="point | line | polygon"**
workspace **="*string*"**

When parent element is ***LAYER and the layer type is image:***
name **="*string*"**
workspace **="*string*"**
type ="image"

When parent element is ***LAYER in GET_IMAGE or GET_EXTRACT:***
fromlayer **="Layer ID in map configuration file"**
>

(m) <PARTITION... />

</DATASET >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Defines the dataset used in the layer.

Restrictions:

- When in a request, can only be used with LAYER in a GET_EXTRACT or GET_IMAGE element.
- In a GET_IMAGE request, use either
  name, *type*, and *workspace*
  or use
  fromlayer
  One of these two attribute sets is required.
- The attribute *type* is required if the layer is a featureclass type (point, line, or polygon). *Type* is not required for raster image layers.
- ArcMap Server supports only the attribute *fromlayer*. References can be made only to layers that already exist in an ArcMap Image Service.

Notes:

- There are five different ways to access images and GRIDs in an Image Service as detailed in the table below.

Image Access Method	Workspace Location	DATASET Layer Name
Specify by name.	Use IMAGEWORKSPACE; attribute *directory* points to location of specified image.	Name of image including its extension.
Use all images in a directory. Images in the same directory automatically tile if they use the same coordinate projection and are drawn when they are within the extent requested.	Use IMAGEWORKSPACE; attribute *directory* points to location of the group of images.	Use *ImageDirectory for the name: *name="*ImageDirectory"*
Use an ArcView GIS image catalog.	Use IMAGEWORKSPACE; attribute *directory* points to location of catalog, not images.	Name of the image catalog DBF file. For instance, if catalog is named imagecat.dbf, use *name="imagecat.dbf"*.
Add a GRID.	Use IMAGEWORKSPACE. A GRID has two directories: one for the GRID data and one for the INFO files. Both these directories should be grouped together under a parent directory. The *directory* attribute points to the	Name of directory that contains GRID data; for a GRID named WorldImage, use *name="WorldImage"*.

	parent directory above the GRID and INFO directories. A *.clr file can be included to color the GRID. This file should have the same name as the GRID and be included in the parent directory above the INFO and GRID directories.	
Use an image in ArcSDE.	Use SDEWORKSPACE.	Name is the full ArcSDE name including the field name where the image resides. In the following examples, "IMAGE" is the field where the image resides: For Oracle, assuming a user of "RASTER" and an image layer named "MYIMAGE", the name is RASTER.MYIMAGE.IMAGE. For SQL Server, Informix, and DB2, assuming a user and database named "RASTER" and an image layer named "MYIMAGE", the name is RASTER.RASTER.MYIMAGE.IMAGE

- For a listing of supported images, see *Using ArcIMS*.
- In a request, when using the *fromlayer* attribute on a layer, if this layer has been constrained with a query in the map configuration file, *fromlayer* inherits this query.

Attribute Descriptions for DATASET:

*When parent element is **LAYER and the layer type is featureclass:***

Attribute	Usage
name	The value depends on whether DATASET is used in a map configuration file or a viewer configuration file. • Map configuration file. For shapefiles, use the name of the data file without an extension such as STATES. For ArcSDE, use the full name of the layer such as DATA.STATES. • Feature Service layers in a viewer configuration file. Use the LAYER *id* in the map configuration file. For example, if the

id for a layer is "0", then DATASET name="0".
- Image Service layers in a viewer configuration file. Use the name of the Image Service. For example, if the service is named "world", then DATASET name="world".

type	Source layer feature type. Required when the layer source defined in LAYER is *featureclass*.
workspace	References the workspace name where the data resides.

*When parent element is **LAYER and the layer type is image:***

Attribute	Usage
name	For images, see the table in the Notes section for details on accessing and naming image files.
type	Layer source type is optional for image layers.
workspace	References the workspace name where the data resides.

*When parent element is **LAYER in GET_IMAGE or GET_EXTRACT:***

Attribute	Usage
fromlayer	References an existing layer in a map configuration file. Use the layer ID, not name.

Examples for DATASET:

Example 1: When used in a map configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.21" miny="18.92" maxx="-66.96"
maxy="71.41" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CITIES" visible="true" id="2">
      <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLEMARKERSYMBOL type="square" width="5" />
```

```
      </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When in a viewer configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-180.0" miny="-90.0" maxx="185.6901927947995"
maxy="90.0" name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
      <IMAGESERVERWORKSPACE name="mapper_ws-0"
url="http://mymachine.domain.com/servlet/com.esri.esrimap.Esrimap"
service="background" />
      <FEATURESERVERWORKSPACE name="ifs_ws-1"
url="http://mymachine.domain.com/servlet/com.esri.esrimap.Esrimap"
service="world" />
      </WORKSPACES>
      <LAYER type="image" name="background" visible="true" id="0">
      <DATASET name="background" type="image" workspace="mapper_ws-0"
/>
      </LAYER>
      <LAYER type="featureclass" name="Countries" visible="true"
id="1">
      <DATASET name="1" type="polygon" workspace="ifs_ws-1" />
      </LAYER>
    </MAP>
    <SCALEBAR backcolor="192,192,192" fontcolor="0,0,0"
mapunits="decimal_degrees" scaleunits="feet" screenunits="inches" />
  </CONFIG>
</ARCXML>
```

Example 3: When using fromlayer in a REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <IMAGESIZE width="643" height="502" />
      </PROPERTIES>
```

```
      <LAYER type="featureclass" name="select layer" visible="true"
id="Selected">
        <DATASET fromlayer="1" />
        <SPATIALQUERY>
          <SPATIALFILTER relation="area_intersection">
            <POLYLINE>
              <PATH>
                <POINT x="-2.1079549837513" y="19.99815142335011" />
                <POINT x="28.99468788980437" y="15.88488432940315" />
                <POINT x="55.99468788980437" y="35.88488432940315" />
              </PATH>
            </POLYLINE>
          </SPATIALFILTER>
        </SPATIALQUERY>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="0,255,0" filltype="cross"
fillinterval="3" />
        </SIMPLERENDERER>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 4: When specifying one image by name.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="192837" miny="3769109" maxx="197809"
maxy="3773771" name="Initial_Extent" />
        <MAPUNITS units="meters" />
      </PROPERTIES>
      <WORKSPACES>
        <IMAGEWORKSPACE directory="<path to image>" name="jai_ws-0" />
      </WORKSPACES>
      <LAYER type="image" name="reno.sid" visible="true" id="0">
        <DATASET name="reno.sid" type="image" workspace="jai_ws-0" />
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 5: When specifying multiple images in a directory.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="0" miny="-1.0" maxx="891.0" maxy="1000.0"
name="Initial_Extent" />
        <MAPUNITS units="feet" />
      </PROPERTIES>
      <WORKSPACES>
        <IMAGEWORKSPACE directory="<path to image directory>"
name="jai_ws-0" />
      </WORKSPACES>
      <LAYER type="image" name="*ImageDirectory" visible="false"
id="0">
        <DATASET name="*ImageDirectory" type="image"
workspace="jai_ws-0" />
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 6: When specifying an image catalog.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="0" miny="0" maxx="2000000" maxy="2000000"
name="Initial_Extent"/>
         <MAPUNITS units="meters" />
      </PROPERTIES>
      <WORKSPACES>
        <IMAGEWORKSPACE name="jai_ws-15" directory="<path to image
catalog dbf file>"/>
      </WORKSPACES>
      <LAYER type="image" name="Mammoth Area" visible="true" id="0">
        <DATASET name="mammoth.dbf" type="image" workspace="jai_ws-15"
/>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 7: When specifying a GRID.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
       <ENVELOPE minx="-2006008" miny="-1240677" maxx="-1993628"
maxy="-1256187" name="Initial_Extent" />
       <MAPUNITS units="meters" />
      </PROPERTIES>
      <WORKSPACES>
       <IMAGEWORKSPACE directory="<path to parent directory above INFO
and GRID directory>" name="jai_ws-15" />
      </WORKSPACES>
      <LAYER type="image" name="Mt St. Helens" visible="true" id="0">
       <DATASET name="<GRID directory>" type="image"
workspace="jai_ws-15" />
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

DELETE_METADATA

Used in: REQUEST
Servers: Metadata (Publish)
Parent elements: PUBLISH_METADATA

<DELETE_METADATA

docid ="*string*"
>

No Child Elements
</DELETE_METADATA >

Bold: Attribute or child element is required.

Description:

Deletes a metadata document from the metadata repository.

Restrictions:

- This element can be used only by the owner established in PUT_METADATA. Any other user will get an error message.

Notes:

- See METADATA_ACTION for response.

Attribute Descriptions for DELETE_METADATA:

Attribute	Usage
docid	String used to uniquely identify a document. The client used to publish the metadata is responsible for creating the document ID. This ID is automatically assigned when using ArcCatalog. If another client is used, GET_UUID can be used to request a valid ID. The format for an ID is the following: {HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH} where H is a hexadecimal digit (0-9,a-f,A-F). The ID is limited to 38 characters.

Examples for DELETE_METADATA:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <PUBLISH_METADATA>
      <DELETE_METADATA docid="{C64D8F38-82B4-11D5-99C2-
000086460FA0}" />
    </PUBLISH_METADATA>
  </REQUEST>
</ARCXML>
```

DELETE_METADATA_RELATIONSHIP

Used in: REQUEST
Servers: Metadata (Publish)
Parent elements: PUBLISH_METADATA

<DELETE_METADATA_RELATIONSHIP >
No Attributes

<METADATA_SOURCE... />
(m) <METADATA_CHILD... /> *[And/Or]*
(m) <METADATA_SIBLING... /> *[And/Or]*

</DELETE_METADATA_RELATIONSHIP >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Deletes relationships beween a source metadata document and one or more child or sibling metadata documents.

Restrictions:

None

Notes:

- See METADATA_ACTION for response.
- A child document is a subdocument to the current metadata document. A sibling document is a related document. When selected as a sibling, the document is listed under the parent document under a link called "related records".

Examples for DELETE_METADATA_RELATIONSHIP:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <PUBLISH_METADATA>
      <DELETE_METADATA_RELATIONSHIP>
        <METADATA_SOURCE docid="{66843298-85C5-11D5-99C3-000086460FA0}"
/>
        <METADATA_CHILD docid="{1F7DDF21-BC01-4C20-8AA5-243B33ED0B1E}"
/>
      </DELETE_METADATA_RELATIONSHIP>
```

```
    </PUBLISH_METADATA>
  </REQUEST>
</ARCXML>
```

DELETEDFEATURES

Used in: MARKUP
Parent elements: MARKUPLAYER

<DELETEDFEATURES

deletedfeatures ***="string"***
>

(m) <FEATURE... />

</DELETEDFEATURES >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Defines features deleted from a specified layer during an EditNotes session.

Restrictions:

None

Notes:

- The *deletedfeatures* attribute contains a list of identifiers for each deleted feature. An identifier in this list must match a *featureid* in one of the FEATURE child elements. For example, if <DELETEDFEATURES deletedfeatures="333">, then <FEATURE featureid="333">.

Attribute Descriptions for DELETEDFEATURES:

Attribute	Usage
deletedfeatures	A list of all deleted feature IDs. ID values are always separated by a comma.

Examples for DELETEDFEATURES:

Example 1:

```
<?xml version="1.0"?>
<MARKUP>
  <WORKSPACES>
    <FEATURESERVERWORKSPACE name="ifs_ws-0"
url="http://mycomputer.domain.com/servlet/com.esri.esrimap.Esrimap"
```

```
service="states" />
  </WORKSPACES>
  <MARKUPLAYER layername="States" workspace="ifs_ws-0">
    <DELETEDFEATURES deletedFeatures="22">
      <FEATURE featureid="22">
      <ENVELOPE minx="-119.99807766185346" miny="34.989467399607044"
maxx="-114.03885920596963" maxy="42.00172044207277" />
      <FIELD name="AREA" precision="3" size="12" type="8">
        <FIELDVALUE valuestring="110667.293" />
      </FIELD>
      <FIELD name="STATE_NAME" precision="0" size="25" type="12">
        <FIELDVALUE valuestring="Nevada" />
      </FIELD>
      <FIELD name="STATE_FIPS" precision="0" size="2" type="12">
        <FIELDVALUE valuestring="32" />
      </FIELD>
      <FIELD name="SUB_REGION" precision="0" size="7" type="12">
        <FIELDVALUE valuestring="Mtn" />
      </FIELD>
      <FIELD name="STATE_ABBR" precision="0" size="2" type="12">
        <FIELDVALUE valuestring="NV" />
      </FIELD>
      <FIELD name="#SHAPE#" precision="0" size="0" type="-98">
        <FIELDVALUE>
          <POLYGON>
            <RING>
            <POINT x="-119.16043651343605" y="38.41182980599817" />
            <POINT x="-119.87841464065097" y="38.91441449504861" />
            ...
            <POINT x="-119.16043651343601" y="38.41182980599817" />
            <POINT x="-119.16043651343605" y="38.41182980599817" />
            </RING>
          </POLYGON>
        </FIELDVALUE>
      </FIELD>
      </FEATURE>
    </DELETEDFEATURES>
  </MARKUPLAYER>
</MARKUP>
```

DENSIFY

Used in: **CONFIG** **REQUEST**
Servers: Image Feature Extract
Parent elements: **LAYER**

<DENSIFY

tolerance **="*double*"**
>

No Child Elements
</DENSIFY >

Bold: Attribute or child element is required.

Description:

The process of data densification adds points to a layer before the layer is projected. DENSIFY sets the interval used for adding points.

Restrictions:

None

Notes:

- Using DENSIFY can slow down the ArcIMS Spatial Server noticeably. Use DENSIFY only when features are missing from one or more layers in a requested image, usually around the periphery. It is better to start with a large number for *tolerance*. A guideline is to start with a value about 20 to 30 percent of the distance between the minimum and maximum x-coordinates of the current map extent.
- The units for *tolerance* are the same as the units for the layer. For example, if the layer is in feet, the *tolerance* distance is in feet.
- DENSIFY is only needed if a layer is going to be projected. If the layer is in the same projection as the ArcIMS service, DENSIFY does not need to be used.
- For more information on the projection elements, see Using Projection Elements.

Attribute Descriptions for DENSIFY:

Attribute	Usage
tolerance	Defines distance (tolerance) between points and is applied on geometry before projecting takes place.

Examples for DENSIFY:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <MAPUNITS units="meters" />
        <FEATURECOORDSYS id="54008" />
        <FILTERCOORDSYS id="4326"  />
      </PROPERTIES>
      <WORKSPACES>
      <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="Cntry94" visible="true" id="0">
        <DATASET name="Cntry94" type="polygon" workspace="shp_ws-0" />
        <COORDSYS id="4326" />
        <DENSIFY tolerance="100" />
        <SIMPLERENDERER>
           <SIMPLEPOLYGONSYMBOL filltransparency="1.0"
fillcolor="27,127,127" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

DOCUMENTINFO

Used in: REQUEST
Servers: Metadata (Browse)
Parent elements: SEARCH_METADATA

```
<DOCUMENTINFO

    name ="string"
    owner ="string"
>

    No Child Elements
</DOCUMENTINFO >
```

Description:

Specifies the name and/or owner of a document allowing you to search for all documents with a specific name or all documents owned by a certain user.

Restrictions:

None

Notes:

None

Attribute Descriptions for DOCUMENTINFO:

Attribute	Usage
name	Name that identifies the dataset corresponding to the document.
owner	Name identifying the owner of the metadata document.

Examples for DOCUMENTINFO:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <SEARCH_METADATA>
        <DOCUMENTINFO name="World" />
      </SEARCH_METADATA>
    </GET_METADATA>
  </REQUEST>
</ARCXML>
```

DRAW

Used in: **REQUEST**
Servers: Image ArcMap
Parent elements: **PROPERTIES**

<DRAW

map ="true | false"
>

No Child Elements
</DRAW >

Bold: Attribute or child element is required.

Description:

Disables map generation when only a legend is needed during a GET_IMAGE request.

Restrictions:

None

Notes:

- For more information on using DRAW in a GET_IMAGE request, see Using GET_IMAGE and IMAGE with Image Services or Using GET_IMAGE and IMAGE with ArcMap Image Services.

Attribute Descriptions for DRAW:

Attribute	Usage
map	Turns map generation on or off.

Examples for DRAW:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <LEGEND title="Legend" font="Arial" autoextend="true"
columns="2" width="170" height="300" backgroundcolor="255,255,255" />
```

```
        <DRAW map="false"/>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

ENVELOPE

Used in: **CONFIG** **REQUEST** **RESPONSE** **MARKUP**
Servers: Image Query Feature Extract Metadata ArcMap
Parent elements: **DATAFRAME** **EXTRACT** **FCLASS** **FEATURE** **FEATURES** **IMAGE** **LAYERINFO** **LAYOUT** **LAYOUTINFO** **METADATA_DATASET** **PARTITION** **PROPERTIES** **PUT_METADATA** **SEARCH_METADATA** **SPATIALFILTER**

<ENVELOPE

*When parent element is **ALL except PROPERTIES and SEARCH_METADATA:***
maxx ="***double***"
maxy ="***double***"
minx ="***double***"
miny ="***double***"

*When parent element is **PROPERTIES:***
maxx ="***double***"
maxy ="***double***"
minx ="***double***"
miny ="***double***"
name ="Initial_Extent | Extent_Limit" **[Initial_Extent]**
reaspect ="true | false" **[true]**

*When parent element is **SEARCH_METADATA:***
maxx ="***double***"
maxy ="***double***"
minx ="***double***"
miny ="***double***"
spatialoperator ="within | overlaps" **[overlaps]**
>

No Child Elements
</ENVELOPE >

Bold: Attribute or child element is required.

Description:

Provides a minimum and maximum x and y extent. In a request, the ENVELOPE extent is usually defined by the user. In a response, the ENVELOPE extent defines the boundary of the returned features. In a configuration file, ENVELOPE defines the initial extent of a map view. When used with metadata, the ENVELOPE defines the geographic extent of a document.

Restrictions:

- *Name* is used only in viewer configuration files, most commonly default.axl, when parent element is PROPERTIES.
- *Reaspect* is used only when parent element is PROPERTIES. It is valid only for Image Services and is ignored in Feature and ArcMap Image Services.

Notes:

- When ENVELOPE is used as a child element to LAYOUTINFO or LAYOUT, the values for the envelope are in page units rather than map units.

Attribute Descriptions for ENVELOPE:

*When parent element is **ALL except PROPERTIES and SEARCH_METADATA:***

Attribute	Usage
maxx	Right top x-coordinate. Units are in map units for all elements except LAYOUT and LAYOUTINFO, which are in page units.
maxy	Right top y-coordinate. Units are in map units for all elements except LAYOUT and LAYOUTINFO, which are in page units.
minx	Left bottom x-coordinate. Units are in map units for all elements except LAYOUT and LAYOUTINFO, which are in page units.
miny	Left bottom y-coordinate. Units are in map units for all elements except LAYOUT and LAYOUTINFO, which are in page units.

*When parent element is **PROPERTIES:***

Attribute	Usage
maxx	Right top x-coordinate in map units.
maxy	Right top y-coordinate in map units.
minx	Left bottom x-coordinate in map units.
miny	Left bottom y-coordinate in map units.
name	Identifies the type of envelope extent in a viewer configuration file. Initial_Extent is the full extent drawn when a file is first accessed. Extent_Limit is the maximum extent allowable when zooming out. When an extent limit is used in a viewer configuration file, two ENVELOPE elements are needed: <PROPERTIES> <ENVELOPE minx="-61.1" miny="3.7" maxx="91.7" maxy="61.3" **name="Initial_Extent"**/> <ENVELOPE minx="-61.1" miny="3.7" maxx="91.7" maxy="61.3" **name="Extent_Limit"**/> <PROPERTIES>

	The two ENVELOPE extents do not need to be the same.
reaspect	Flag indicating whether the ENVELOPE should be stretched to fit the viewing area in the client. By default, the pixel width and height ratio stays the same. By setting *reaspect* to false, the pixel width and height are stretched. Valid only with Image Services.

*When parent element is **SEARCH_METADATA:***

Attribute	Usage
maxx	Right top x-coordinate in map units.
maxy	Right top y-coordinate in map units.
minx	Left bottom x-coordinate in map units.
miny	Left bottom y-coordinate in map units.
spatialoperator	The restrictions on the envelope.

Examples for ENVELOPE:

Example 1: When limiting a metadata search to the region defined within the ENVELOPE.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <SEARCH_METADATA operator="and">
        <ENVELOPE xmin="-176.98" ymin="18.93" xmax="-66.97"
ymax="71.35" spatialoperator="within" />
        <TAGVALUE
tag="metadata/dataqual/lineage/srcinfo/srccite/citeinfo/pubdate"
greaterthan="1990" />
        <FULLTEXT word="cities"/>
        <SEARCH_METADATA operator="or">
          <TAGTEXT tag="metadata/idinfo/citation/citeinfo/geoform"
word="data"/>
          <TAGTEXT tag="metadata/idinfo/citation/citeinfo/geoform"
word="digital"/>
        </SEARCH_METADATA>
      </SEARCH_METADATA>
    </GET_METADATA>
  </REQUEST>
</ARCXML>
```

Example 2: Used in CONFIG to set the initial extent.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
```

```
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
       <ENVELOPE minx="-178.2" miny="18.9" maxx="-66.9" maxy="71.4"
name="Initial_Extent" />
       <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
       <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="STATES" visible="true" id="0">
       <DATASET name="STATES" type="polygon" workspace="shp_ws-0" />
       <SIMPLERENDERER>
         <SIMPLEPOLYGONSYMBOL fillcolor="255,255,153" filltype="solid"
/>
       </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 3: When in a GET_IMAGE REQUEST to set the selection area for the spatial filter (first ENVELOPE) and the overall extent of the map (second ENVELOPE).

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
  <GET_IMAGE>
    <PROPERTIES>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      <LAYERLIST>
        <LAYERDEF id="Cities">
          <SIMPLERENDERER>
            <SIMPLEMARKERSYMBOL size="16" color="0,0,0" />
          </SIMPLERENDERER>
         <SPATIALQUERY>
            <SPATIALFILTER relation="area_intersection">
            <ENVELOPE maxy="60" maxx="60" miny="0" minx="0" />
         </SPATIALFILTER>
         </SPATIALQUERY>
      </LAYERDEF>
      </LAYERLIST>
   </PROPERTIES>
  </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 4: Identifies the initial extent and the extent of a layer in a SERVICEINFO response.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="Cities"
id="0">
        <FCLASS type="point">
          <ENVELOPE minx="-165.270004272461" miny="-53.1500015258789"
maxx="177.130187988281" maxy="78.1999969482422" />
        </FCLASS>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Example 5: When using in GET_LAYOUT. Note the ENVELOPE in the PROPERTIES section is in page units.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_LAYOUT>
      <PROPERTIES>
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
        <FILTERCOORDSYS id="54030" />
        <FEATURECOORDSYS id="54030" />
        <IMAGESIZE width="800" height="600" />
        <OUTPUT type="JPG" />
      </PROPERTIES>
      <DATAFRAME id="Layers" >
        <FILTERCOORDSYS id="4326" />
        <FEATURECOORDSYS id="4326" />
        <ENVELOPE minx="-121" miny="36" maxx="-112" maxy="44" />
      </DATAFRAME>
    </GET_LAYOUT>
  </REQUEST>
</ARCXML>
```

ENVIRONMENT

Used in: CONFIG REQUEST RESPONSE Application Server RESPONSE
Parent elements: CONFIG GET_EXTRACT GET_FEATURES GET_IMAGE SERVICE SERVICEINFO

<ENVIRONMENT >
No Attributes

*When parent element is **CONFIG:***
<LOCALE... />
<UIFONT... />
<SCREEN... />
<SEPARATORS... />

*When parent element is **GET_EXTRACT, GET_FEATURES, GET_IMAGE:***
<SEPARATORS... />

*When parent element is **SERVICE:***
<LOCALE... />
<UIFONT... />

*When parent element is **SERVICEINFO:***
<IMAGELIMIT... />
<LOCALE... />
<SCREEN... />
<SEPARATORS... />
<UIFONT... />
<CAPABILITIES... />

</ENVIRONMENT >

Bold: Attribute or child element is required.

Description:

The parent element for setting the ArcIMS environment.

Restrictions:

- IMAGELIMIT and SCREEN are not included in a SERVICEINFO response when the request has been rerouted to the Geocode Server.

Notes:

None

Examples for ENVIRONMENT:

Example 1: When in CONFIG.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180,0" miny="-90,0" maxx="180,0" maxy="90,0"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-18" directory="<path to WORLD
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="0">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-18" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="127,227,127" filltype="solid"
/>
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When in an application server RESPONSE to GETCLIENTSERVICES.

```
<?xml version="1.0"?>
<ARCXML version="1.1">
<RESPONSE>
  <SERVICES>
    <SERVICE name="world_image" servicegroup="ImageServer1"
access="PUBLIC" type="ImageServer" desc=""  status="ENABLED" >
      <ENVIRONMENT>
        <LOCALE country="US" language="en" />
        <UIFONT name="Arial" />
      </ENVIRONMENT>
      <CLEANUP interval="20" />
    </SERVICE>
  </SERVICES>
</RESPONSE>
</ARCXML>
```

ERROR

Used in: **RESPONSE**
Servers: Image Query Feature Extract Geocode ArcMap
Parent elements: **RESPONSE**

```
<ERROR

    machine ="string"
    processid ="integer"
    threadid ="integer"
>

    No Child Elements
</ERROR >
```

Bold: Attribute or child element is required.

Description:

Contains an error message from the ArcIMS Spatial Server.

Restrictions:

None

Notes:

Attribute Descriptions for ERROR:

Attribute	Usage
machine	Machine name where the ArcIMS Spatial Server processing the request resides.
processid	Unique identifier for the ArcIMS Spatial Server that processed the request.
threadid	ArcIMS Spatial Server instance (thread) processing the request.

Examples for ERROR:

Example 1:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <ERROR machine="REYKJAVIK" processid="440" threadid="398">
```

```
      AXLParser ERROR: 'minx' not found in ENVELOPE
    </ERROR>
  </RESPONSE>
</ARCXML>
```

EXACT

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Feature
Parent elements: **VALUEMAPLABELRENDERER** **VALUEMAPRENDERER**

<EXACT

label ="*string*"
method ="IsExact | isContained" **[IsExact]**
value ="*string, numeric, or date*"
>

*When parent element is **VALUEMAPLABELRENDERER:***
<CALLOUTMARKERSYMBOL... /> *[Or]*
<CHARTSYMBOL... /> *[Or]*
<RASTERSHIELDSYMBOL... /> *[Or]*
<SHIELDSYMBOL... /> *[Or]*
<TEXTSYMBOL... /> *[Or]*

*When parent element is **VALUEMAPRENDERER:***
<GRADIENTFILLSYMBOL... /> *[Or]*
<HASHLINESYMBOL... /> *[Or]*
<RASTERFILLSYMBOL... /> *[Or]*
<RASTERMARKERSYMBOL... /> *[Or]*
<SIMPLELINESYMBOL... /> *[Or]*
<SIMPLEMARKERSYMBOL... /> *[Or]*
<SIMPLEPOLYGONSYMBOL... /> *[Or]*
<TRUETYPEMARKERSYMBOL... /> *[Or]*

</EXACT >

Bold: Attribute or child element is required.

Description:

Used with VALUEMAPRENDERER and VALUEMAPLABELRENDERER for matching exact values within a specified field in the database. When a match occurs, the symbol is drawn as specified for that EXACT value.

Restrictions:

- One renderer child element is required for each EXACT element in the value map.
- Not valid with ArcMap Server.

Notes:

- If DATASET layer type is point, then only point symbols can be used. If type is line, then line and point symbols can be used. If type is polygon, then polygon, line, and point symbols can be used.
- If there are leading or trailing blanks in a field value, they will be trimmed before a comparison is made. For example, a field value of " Hello " is interpreted as "Hello".
- For more information on using renderers, see Using ArcXML Renderers.

Attribute Descriptions for EXACT:

Attribute	Usage
label	Label for legend.
method	Refers to the way a value in the data field is compared to the EXACT value. Use isExact for an exact match. Use isContained to search for the value anywhere in a string. String comparisons are case sensitive.
value	Values used for matching records in a selected field. They can be a numeric, string, or date value. Multiple values can be grouped together for one EXACT category. The default separator between values is a semicolon. The separator can be changed by declaring the tuple separator (*ts*) in SEPARATORS.

Examples for EXACT:

Example 1: When using VALUEMAPRENDERER.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="477518.305458" miny="3762433.842048"
maxx="495210.599161" maxy="3773575.120050" name="Initial_Extent" />
         <MAPUNITS units="meters" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to data>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="Crime" visible="true" id="2">
        <DATASET name="Crime" type="point" workspace="shp_ws-0" />
        <VALUEMAPRENDERER lookupfield="CODE">
        <EXACT value="3" label="Level 3">
          <SIMPLEMARKERSYMBOL color="127,27,27" type="cross" width="6"
/>
```

```
        </EXACT>
        <EXACT value="2" label="Level 2">
          <SIMPLEMARKERSYMBOL color="227,27,27" type="triangle"
width="6" />
        </EXACT>
        <EXACT value="1" label="Level 1">
          <TRUETYPEMARKERSYMBOL font="ESRI Cartography" character="252"
fontcolor="255,255,0" fontsize="16" />
        </EXACT>
        </VALUEMAPRENDERER>
      </LAYER>

    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When using VALUEMAPLABELRENDERER.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.215027" miny="18.924782" maxx="-66.969849"
maxy="71.406647" name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="ROADS" visible="true" id="1">
        <DATASET name="ROADS" type="line" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <VALUEMAPLABELRENDERER lookupfield="ADMN_CLASS"
labelfield="ROUTE" linelabelposition="placeabove">
          <EXACT value="US Highway" label="US Highways"
method="isContained" >
            <SHIELDSYMBOL antialiasing="true" font="Arial"
fontstyle="regular" fontsize="10" type="usroad" />
          </EXACT>
          <EXACT value="Interstate" label="Interstate Freeways"
method="isContained" >
            <SHIELDSYMBOL labelmode="numericonly" antialiasing="true"
font="Tahoma" fontstyle="italic" fontsize="14" type="interstate"
minsize="1"/>
          </EXACT>
          <OTHER>
             <TEXTSYMBOL font="Arial" fontstyle="regular" fontsize="10"
```

```
/>
          </OTHER>
          </VALUEMAPLABELRENDERER>
          <SIMPLERENDERER>
            <SIMPLELINESYMBOL type="solid" width="1" captype="round"
jointype="round" color="127,127,27" />
          </SIMPLERENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 3: Example showing multiple values grouped together for one EXACT category. The separator, by default, is a semicolon. The separator can be changed by changing the tuple separator (ts) in SEPARATORS.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-178.215027" miny="18.924782" maxx="-66.969849"
maxy="71.406647" name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
      <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="STATES" visible="true" id="0">
      <DATASET name="STATES" type="polygon" workspace="shp_ws-0" />
      <VALUEMAPRENDERER lookupfield="STATE_ABBR">
        <EXACT value="OK;OH;OR" label="States with O">
          <SIMPLEPOLYGONSYMBOL fillcolor="27,27,27" filltype="solid" />
        </EXACT>
        <EXACT value="NV;NE;ND;NC;NY;NH;NJ;NM" label="States with N">
          <SIMPLEPOLYGONSYMBOL fillcolor="27,27,127" filltype="solid"
/>
        </EXACT>
        <EXACT value="MT;ME;MA;MD;MS;MO;MI;MN" label="States with M">
          <SIMPLEPOLYGONSYMBOL fillcolor="227,227,227" filltype="solid"
/>
        </EXACT>
        <OTHER>
          <SIMPLEPOLYGONSYMBOL fillcolor="127,27,27" filltype="solid"
/>
        </OTHER>
      </VALUEMAPRENDERER>
      </LAYER>
```

```
    </MAP>
  </CONFIG>
</ARCXML>
```

EXTENSION

Used in: **CONFIG** **RESPONSE**
Servers: Image Query Feature Extract Geocode
Parent elements: **LAYER** **LAYERINFO**

<EXTENSION

type ="Geocode | StoredQuery | Extract"
>

<EXTRACTPARAMS... /> *[Or]*
<GCSTYLE... /> *[Or]*
(m) <STOREDQUERIES... /> *[Or]*

</EXTENSION >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Defines geocoding, stored queries, or extract parameters for a layer.

Restrictions:

None

Notes:

- Use GCSTYLE for a geocode extension, STOREDQUERIES for a stored query extension, and EXTRACTPARAMS to set up parameters for the Extract Server. Only one of the child elements can be used within an EXTENSION. Multiple EXTENSION elements can be used within a layer.

Attribute Descriptions for EXTENSION:

Attribute	Usage
type	Extension type: Geocode, StoredQuery, or Extract.

Examples for EXTENSION:

Example 1: Describes a layer with two extensions: one for geocoding and another for stored queries.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
       <ENVELOPE minx="-71.077092" miny="42.357962" maxx="-71.034511"
maxy="42.385263" name="Initial_Extent" />
       <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
       <SHAPEWORKSPACE name="shp_ws-64" directory="<path to data>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="Streets" visible="true" id="4">
       <DATASET name="bosstreets" type="line" workspace="shp_ws-64" />
       <SIMPLERENDERER>
         <SIMPLELINESYMBOL type="solid" width="2" color="255,0,0" />
       </SIMPLERENDERER>
       <EXTENSION type="Geocode">
         <GCSTYLE name="USAddressZ">
           <GCFIELD id="FromLeft" name="L_F_ADD" />
           <GCFIELD id="FromRight" name="R_F_ADD" />
           <GCFIELD id="ToLeft" name="L_T_ADD" />
           <GCFIELD id="ToRight" name="R_T_ADD" />
           <GCFIELD id="PreDir" name="PREFIX" />
           <GCFIELD id="PreType" name="PRE_TYPE" />
           <GCFIELD id="StreetName" name="NAME" />
           <GCFIELD id="StreetType" name="TYPE" />
           <GCFIELD id="SufDir" name="SUFFIX" />
           <GCFIELD id="LeftZone" name="ZIPL" />
           <GCFIELD id="RightZone" name="ZIPR" />
         </GCSTYLE>
       </EXTENSION>
       <EXTENSION type="StoredQuery">
         <STOREDQUERIES>
           <STOREDQUERY name="Streets">
             <QUERY where=" NAME  = &apos;[%var%]&apos;"
subfields="#SHAPE# L_F_ADD L_T_ADD R_F_ADD R_T_ADD PREFIX PRE_TYPE NAME
TYPE SUFFIX ZIPL ZIPR CITYL CITYR STATE_ABBR CFCC ROAD_TYPE" />
             <SQVAR position="0" name="[%var%]">
              <FIELD name="NAME" precision="0" type="12" size="32" />
             </SQVAR>
           </STOREDQUERY>
         </STOREDQUERIES>
       </EXTENSION>
      </LAYER>
    </MAP>
```

```
  </CONFIG>
</ARCXML>
```

Example 2: When using an EXACTPARAMS extension.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP dynamic="true">
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-2" directory="<path to USA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="STATES" visible="true"
id="States">
        <DATASET name="STATES" type="polygon" workspace="shp_ws-2" />
        <EXTENSION type="extract" >
          <EXTRACTPARAMS clip="true" />
        </EXTENSION>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="CITIES" visible="true" id="35">
        <DATASET name="CITIES" type="point" workspace="shp_ws-2" />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="102,0,102" width="8.0" />
        </SIMPLERENDERER>
        <SPATIALQUERY where="POP1990 &gt; 100000" subfields="CITY_NAME
STATE_NAME POP1990 MALES FEMALES"  />
        <EXTENSION type="extract" >
        <EXTRACTPARAMS clip="true" >
          <OUTPUTFILE file="us_cities" >
            <OUTPUTFIELD name="CITY_NAME" alias="City" />
            <OUTPUTFIELD name="STATE_NAME" alias="State" />
            <OUTPUTFIELD name="POP1990" alias="Population" />
            <OUTPUTFIELD name="MALES" alias="Male_pop" />
            <OUTPUTFIELD name="FEMALES" alias="Female_pop" />
          </OUTPUTFILE>
        </EXTRACTPARAMS>
        </EXTENSION>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

EXTRACT

Used in: RESPONSE
Servers: Extract
Parent elements: RESPONSE

<EXTRACT >
No Attributes

<ENVELOPE... />
<OUTPUT... />

</EXTRACT >

Bold: Attribute or child element is required.

Description:

Provides the name and location of the ZIP file containing shapefiles.

Restrictions:

None

Notes:

- See GET_EXTRACT for request.
- See Using GET_EXTRACT and EXTRACT for more information and additional examples.

Examples for EXTRACT:

Example 1:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <EXTRACT>
      <ENVELOPE minx="-118.0" miny="34.0" maxx="-117.0" maxy="35.0" />
      <OUTPUT file="c:\arcims\output\mymapservice_mymachine4982332.zip"
url="http://mymachine.domain.com/mymapservice_mymachine4982332.zip" />
    </EXTRACT>
  </RESPONSE>
</ARCXML>
```

EXTRACTPARAMS

Used in: **CONFIG** **RESPONSE**
Servers: Extract
Parent elements: **EXTENSION**

```
<EXTRACTPARAMS

    clip ="true | false"  [false]
>

    <OUTPUTFILE... />

</EXTRACTPARAMS >
```

Description:

Provides the structure for setting up parameters for shapefiles generated with the Extract Server.

Restrictions:

None

Notes:

- In order to allow GET_EXTRACT requests on an ArcIMS service, the extract EXTENSION must be included in the map configuration file with at least one layer. As part of the extension, EXTRACTPARAMS is required and OUTPUTFILE and OUTPUTFIELD are optional.
- In order to allow clipping for a layer in a map configuration file, EXTENSION and EXTRACTPARAMS must be included, and *clip* must be set to "true".
- For more information on using the Extract Server, see Using GET_EXTRACT and EXTRACT.

Attribute Descriptions for EXTRACTPARAMS:

Attribute	Usage
clip	Determines whether features are clipped at the current extent boundary in the viewer. If no clipping is used, features partially within the envelope are extracted in their entirety. If clipping is used, features are extracted based on the current extent boundary. Note that none of a layer's database values are prorated for any clipped features.

Examples for EXTRACTPARAMS:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP dynamic="true">
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-2" directory="<path to USA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="STATES" visible="true"
id="States">
        <DATASET name="STATES" type="polygon" workspace="shp_ws-2" />
        <EXTENSION type="extract" >
          <EXTRACTPARAMS clip="true" />
        </EXTENSION>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="CITIES" visible="true" id="35">
        <DATASET name="CITIES" type="point" workspace="shp_ws-2" />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="102,0,102" width="8.0" />
        </SIMPLERENDERER>
        <SPATIALQUERY where="POP1990 &gt; 100000" subfields="CITY_NAME
STATE_NAME POP1990 MALES FEMALES"  />
        <EXTENSION type="extract" >
        <EXTRACTPARAMS clip="true" >
          <OUTPUTFILE file="us_cities" >
            <OUTPUTFIELD name="CITY_NAME" alias="City" />
            <OUTPUTFIELD name="STATE_NAME" alias="State" />
            <OUTPUTFIELD name="POP1990" alias="Population" />
            <OUTPUTFIELD name="MALES" alias="Male_pop" />
            <OUTPUTFIELD name="FEMALES" alias="Female_pop" />
          </OUTPUTFILE>
        </EXTRACTPARAMS>
        </EXTENSION>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

FCLASS

Used in: RESPONSE
Servers: Image Query Feature Extract ArcMap
Parent elements: LAYERINFO

<FCLASS

type ="point | line | polygon"
>

When using all servers except ArcMap Server:
<ENVELOPE... />
(m) <FIELD... />

When using ArcMap Server:
<ENVELOPE... />
(m) <FIELD... />
<TOC... />

</FCLASS >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Contains feature class information for a layer.

Restrictions:

None

Notes:

None

Attribute Descriptions for FCLASS:

Attribute	Usage
type	Feature type of layer in map configuration file.

Examples for FCLASS:

Example 1: When in a SERVICEINFO response.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="Countries"
id="1">
        <FCLASS type="polygon">
          <FIELD name="AREA" type="8" size="12" precision="3" />
          <FIELD name="NAME" type="12" size="40" precision="0" />
          <FIELD name="ABBREVNAME" type="12" size="12" precision="0" />
          <FIELD name="FIPS_CODE" type="12" size="2" precision="0" />
          <FIELD name="WB_CNTRY" type="12" size="3" precision="0" />
          <FIELD name="HYPERLINK" type="12" size="60" precision="0" />
          <FIELD name="#SHAPE#" type="-98" size="0" precision="0" />
          <FIELD name="#ID#" type="-99" size="16" precision="0" />
        </FCLASS>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

FEATURE

Used in: **RESPONSE** **MARKUP**
Servers: Query Feature Geocode ArcMap
Parent elements: **ADDEDFEATURES** **DELETEDFEATURES** **FEATURES** **GEOCODE** **MODIFIEDFEATURES**

<FEATURE

*When parent element is **FEATURES, ADDEDFEATURES, MODIFIEDFEATURES:***
No Attributes

*When parent element is **GEOCODE, DELETEDFEATURES:***
featureid ="*integer*"
>

*When parent element is **DELETEDFEATURES, ADDEDFEATURES, MODIFIEDFEATURES:***
<ENVELOPE... />
(m) <FIELD... />

*When parent element is **FEATURES:***
<FIELDS... />
<MULTIPOINT... /> *[Or]*
<POLYGON... /> *[Or]*
<POLYLINE... /> *[Or]*
<ENVELOPE... />

*When parent element is **GEOCODE:***
(m) <FIELD... />

</FEATURE >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

When in a RESPONSE, displays geocoding or query results. When in MARKUP, describes features in a MARKUP report of the EditNotes tool.

Restrictions:

- When parent element is FEATURES, either MULTIPOINT, POLYGON, or POLYLINE, if requested, is returned in addition to FIELDS.

Notes:

None

Attribute Descriptions for FEATURE:

*When parent element is **FEATURES, ADDEDFEATURES, MODIFIEDFEATURES:***
No Attributes

*When parent element is **GEOCODE, DELETEDFEATURES:***

Attribute	Usage
featureid	Provides a unique value for a feature. When in GEOCODE, *featureid* is assigned to each result sequentially. When in DELETEDFEATURES, *featureid* must match one of the values in the deletedfeatures list.

Examples for FEATURE:

Example 1: When used in MARKUP.

```
<?xml version="1.0"?>
<MARKUP>
  <WORKSPACES>
    <FEATURESERVERWORKSPACE name="ifs_ws-0"
url="http://mycomputer.domain.com/servlet/com.esri.esrimap.Esrimap"
service="states" />
  </WORKSPACES>
  <MARKUPLAYER layername="States" workspace="ifs_ws-0">
    <DELETEDFEATURES deletedFeatures="22">
      <FEATURE featureid="22">
      <ENVELOPE minx="-119.99807766185346" miny="34.989467399607044"
maxx="-114.03885920596963" maxy="42.00172044207277" />
      <FIELD name="AREA" precision="3" size="12" type="8">
        <FIELDVALUE valuestring="110667.293" />
      </FIELD>
      <FIELD name="STATE_NAME" precision="0" size="25" type="12">
        <FIELDVALUE valuestring="Nevada" />
      </FIELD>
      <FIELD name="STATE_FIPS" precision="0" size="2" type="12">
        <FIELDVALUE valuestring="32" />
      </FIELD>
      <FIELD name="SUB_REGION" precision="0" size="7" type="12">
        <FIELDVALUE valuestring="Mtn" />
      </FIELD>
      <FIELD name="STATE_ABBR" precision="0" size="2" type="12">
        <FIELDVALUE valuestring="NV" />
      </FIELD>
      <FIELD name="#SHAPE#" precision="0" size="0" type="-98">
        <FIELDVALUE>
          <POLYGON>
            <RING>
            <POINT x="-119.16043651343605" y="38.41182980599817" />
            <POINT x="-119.87841464065097" y="38.91441449504861" />
            ...
```

```
            <POINT x="-119.16043651343601" y="38.41182980599817" />
            <POINT x="-119.16043651343605" y="38.41182980599817" />
            </RING>
          </POLYGON>
        </FIELDVALUE>
      </FIELD>
      </FEATURE>
    </DELETEDFEATURES>
  </MARKUPLAYER>
</MARKUP>
```

Example 2: When used in a FEATURES response.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
<RESPONSE>
  <FEATURES>
    <FEATURE>
      <FIELDS CUST_ID="4" NAME="Customer  4"  #SHAPE#="[Geometry]"
#ID#="3" />
      <MULTIPOINT>
        <POINT x="-122.406680" y="37.747422" />
      </MULTIPOINT>
    </FEATURE>
    <FEATURECOUNT count="1" hasmore="false" />
  </FEATURES>
</RESPONSE>
</ARCXML>
```

FEATURECOORDSYS

Used in: **CONFIG** **REQUEST**
Servers: Image Query Feature Extract Geocode ArcMap
Parent elements: **DATAFRAME** **GET_GEOCODE** **PROPERTIES** **QUERY** **SPATIALQUERY**

<FEATURECOORDSYS

When using ArcMap Server:
id **="*integer*"**
string **="*string*"**

When using Image, Extract, Query, or Feature Server:
id **="*integer*"**
string **="*string*"**
datumtransformid ="*integer*"
datumtransformstring ="*string*"
>

No Child Elements
</FEATURECOORDSYS >

Bold: Attribute or child element is required.

Description:

Projection coordinate system to which layers in an ArcIMS service are projected.

Restrictions:

- Must use either *id* or *string*, but not both.
- For datum transformations either *datumtransformid* or *datumtransformstring* is used, but not both.
- In ArcMap Image Services, *datumtransformid* or *datumtransformstring* are not valid.

Notes:

- For a complete list of supported IDs and definition strings, see:
  - Projected Coordinate Systems Listing [Sorted by projection ID] [Sorted by name]
  - Geographic Coordinate Systems Listing [Sorted by projection ID] [Sorted by name]
  - Datum Transformation Listing [Sorted by projection ID] [Sorted by name]

- When using definition strings, the quotes in the string must be changed to " so the ArcIMS Spatial Server can interpret the string correctly. For example, the definition string for World Mollweide is:

PROJCS["World_Mollweide",GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",0.0174532925199433295]],PROJECTION["Mollweide"],PARAMETER["False_Easting",0],PARAMETER["False_Northing",0],PARAMETER["Central_Meridian",0],UNIT["Meter",1]]

Once the quotes have been changed, the string looks like this:

PROJCS["World_Mollweide",GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",0.0174532925199433295]],PROJECTION["Mollweide"],PARAMETER["False_Easting",0],PARAMETER["False_Northing",0],PARAMETER["Central_Meridian",0],UNIT["Meter",1]]

- The attributes *datumtransformid* and *datumtransformstring* are used when datum transformation information needs to be included. Only datum transformations to and from WGS 1984 are supported.
    - When these attributes are used with COORDSYS and FILTERCOORDSYS, the datum transformation is from a non-WGS 1984 datum to WGS 1984. For example, Pulkovo_1942_To_WGS_1984 (datumtransformid="8157") transforms data from Pulkovo 1942 to WGS 1984.
    - When these attributes are used with FEATURECOORDSYS, the datum transformation is from WGS 1984 to a non-WGS 1984 datum. In the above example, the datum transformation is from WGS 1984 to Pulkovo 1942.

- If FEATURECOORDSYS is used, then FILTERCOORDSYS also needs to be present.
- If a service with FEATURECOORDSYS is used in the ArcIMS HTML Viewer, ArcIMS Java Standard or Custom Viewers, or ArcExplorer 4, FEATURECOORDSYS and FILTERCOORDSYS must use the same value for *id* or *string*.
- When in PROPERTIES, FEATURECOORDSYS defines the default coordinate system for all layers. Using FEATURECOORDSYS in SPATIALQUERY overrides the information in PROPERTIES.
- When used as a child element of SPATIALQUERY in a GET_FEATURES request, it applies only to the layer in that request.
- If a layer does not project, double check that a *.prj file or ArcSDE spatial reference table exists for the layer. If not, COORDSYS must be included with the layer.
- For more information on the projection elements, see Using Projection Elements.

Attribute Descriptions for FEATURECOORDSYS:

Attribute	Usage
datumtransformid	Datum transformation ID. Use either *datumtransformid* or *datumstransformstring*, but not both.
datumtransformstring	Datum transformation definition string. Use either *datumtransformid* or *datumstransformstring*, but not both.
id	Projected or geographic coordinate system ID. Use either *id* or *string*, but not both.
string	Projected or geographic coordinate system definition string. Use either *id* or *string*, but not both.

Examples for FEATURECOORDSYS:

Example 1: Using a coordinate system ID in a map configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <MAPUNITS units="meters" />
        <FEATURECOORDSYS id="54008" />
        <FILTERCOORDSYS id="4326"  />
      </PROPERTIES>
      <WORKSPACES>
      <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="Cntry94" visible="true" id="0">
        <DATASET name="Cntry94" type="polygon" workspace="shp_ws-0" />
        <COORDSYS id="4326" />
        <SIMPLERENDERER>
           <SIMPLEPOLYGONSYMBOL filltransparency="1.0"
fillcolor="27,127,127" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: Using a coordinate system definition string.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-15066114" miny="-9182334" maxx="15311928"
maxy="8551044" />
        <MAPUNITS units="meters" />
        <FEATURECOORDSYS
string="PROJCS[&quot;World_Mollweide&quot;,GEOGCS[&quot;GCS_WGS_1984&qu
ot;,DATUM[&quot;D_WGS_1984&quot;,SPHEROID[&quot;WGS_1984&quot;,6378137,
298.257223563]],PRIMEM[&quot;Greenwich&quot;,0],UNIT[&quot;Degree&quot;
,0.0174532925199433295]],PROJECTION[&quot;Mollweide&quot;],PARAMETER[&qu
ot;False_Easting&quot;,0],PARAMETER[&quot;False_Northing&quot;,0],PARAM
ETER[&quot;Central_Meridian&quot;,0],UNIT[&quot;Meter&quot;,1]]" />
        <FILTERCOORDSYS id="54008"/>
      </PROPERTIES>
      <WORKSPACES>
      <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="Cntry94" visible="true" id="0">
        <DATASET name="Cntry94" type="polygon" workspace="shp_ws-0" />
        <COORDSYS id="4326" />
        <SIMPLERENDERER>
           <SIMPLEPOLYGONSYMBOL filltransparency="1.0"
fillcolor="27,127,127" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 3: Using a datum transformation from WGS 1984 to Pulkovo 1942 (datumtransformid="8206").

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="25.2483" miny="45.6663"
maxx="25.49986079043528" maxy="45.833288705864874"
name="Initial Extent" />
```

```
        <MAPUNITS units="decimal_degrees" />
        <FEATURECOORDSYS id="4284" datumtransformid="8206" />
        <FILTERCOORDSYS id="4326" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-2" directory="<path to data>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="w84shp" visible="true" id="0">
        <DATASET name="w84shp" type="point" workspace="shp_ws-2" />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="0,227,27" width="10" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="pulk42" visible="true" id="1">
        <DATASET name="pulk42" type="point" workspace="shp_ws-2" />
        <COORDSYS id="4284" datumtransformid="8206" />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="0,0,255" width="6" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 4: When in a GET_IMAGE request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-145.0" miny="-30.0" maxx="-125.0" maxy="45.0"
/>
        <IMAGESIZE width="800" height="600" />
        <FEATURECOORDSYS id="54030"  />
        <FILTERCOORDSYS id="4326" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 5: When in a GET_FEATURES request.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES featurelimit="25" beginrecord="0" outputmode="xml"
geometry="true" envelope="true" compact="true">
      <LAYER id="4" />
      <SPATIALQUERY subfields="#ALL#" where="POPULATION > 10000000" >
        <FEATURECOORDSYS id="53030" />
        <SPATIALFILTER relation="area_intersection">
          <ENVELOPE minx="-129" miny="16" maxx="-50" maxy="62"/>
        </SPATIALFILTER>
      </SPATIALQUERY>
```

```
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

FEATURECOUNT

Used in: **RESPONSE**
Servers: Query Feature ArcMap
Parent elements: **FEATURES**

<FEATURECOUNT

count ="***integer***"
hasmore ="**true | false**"
>

No Child Elements
</FEATURECOUNT >

Bold: Attribute or child element is required.

Description:

Contains number of features found in a FEATURES response.

Restrictions:

None

Notes:

- For some additional examples of FEATURECOUNT, see Using GET_FEATURES and FEATURES.

Attribute Descriptions for FEATURECOUNT:

Attribute	Usage
count	Number of features found.
hasmore	Flag for whether any more features need to be returned (true) or if all features have been returned (false).

Examples for FEATURECOUNT:

Example 1:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
<RESPONSE>
  <FEATURES>
    <FEATURE>
      <FIELDS CUST_ID="4" NAME="Customer  4"  #SHAPE#="[Geometry]"
#ID#="3" />
```

```
      <MULTIPOINT>
        <POINT x="-122.406680" y="37.747422" />
      </MULTIPOINT>
    </FEATURE>
    <FEATURECOUNT count="1" hasmore="false" />
  </FEATURES>
</RESPONSE>
</ARCXML>
```

FEATURES

Used in: RESPONSE
Servers: Query Feature ArcMap
Parent elements: RESPONSE

<FEATURES >
No Attributes

(m) <FEATURE... />
<FEATURECOUNT... />
<ENVELOPE... />

</FEATURES >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Returns attribute data, feature geometry data, or both. Data is returned in a binary stream from the Feature Server or in ArcXML format from the Query Server.

Restrictions:

- Used only with Feature, Query, or ArcMap Servers.

Notes:

- See GET_FEATURES for request.
- If *outputmode* in the request is binary, the response is a binary stream. If *outputmode* is xml or newxml, the response is in ArcXML. *Outputmode* should be set to "xml" or "newxml" for Image and ArcMap Image Services.
- In ArcXML mode, feature geometry is returned only if the #shape# field is included in the request. The geometry can be returned in a compact or noncompact format also based on the request.
- ENVELOPE is used two ways in a FEATURES response:
  - As the extent for each feature returned.
  - As the extent of all the features returned in the response.

- For more details on using GET_FEATURES and FEATURES including additional examples, see Using GET_FEATURES and FEATURES.

Examples for FEATURES:

Example 1: When envelope and globalenvelope are both requested.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <FEATURES>
      <FEATURE>
        <ENVELOPE minx="-124.731422424316" miny="45.5432510375977"
maxx="-116.918151855469" maxy="49.0000038146973"/>
      </FEATURE>
      <FEATURE>
        <ENVELOPE minx="-124.558395385742" miny="41.9877891540527"
maxx="-116.469444274902" maxy="46.2362594604492"/>
      </FEATURE>
      <FEATURE>
        <ENVELOPE minx="-124.391471862793" miny="32.5357246398926"
maxx="-114.124450683594" maxy="42.0023460388184"/>
      </FEATURE>
      <FEATURE>
        <ENVELOPE minx="-160.245178222656" miny="18.9247817993164"
maxx="-154.793869018555" maxy="22.2324924468994"/>
      </FEATURE>
      <FEATURE>
        <ENVELOPE minx="-178.215026855469" miny="51.5844345092773"
maxx="-129.990539550781" maxy="71.4066467285156"/>
      </FEATURE>
      <FEATURECOUNT count="5" hasmore="false" />
      <ENVELOPE minx="-178.215026855469" miny="18.9247817993164"
maxx="-114.124450683594" maxy="71.4066467285156"/>
    </FEATURES>
  </RESPONSE>
</ARCXML>
```

Example 2: When attributes are included.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <FEATURES>
      <FEATURE>
        <ENVELOPE minx="-99.127571105957" miny="19.4270458221436"
maxx="-99.127571105957" maxy="19.4270458221436"/>
        <FIELDS NAME="Mexico City" POPULATION="14100000"
#SHAPE#="[Geometry]" />
      </FEATURE>
      <FEATURE>
        <ENVELOPE minx="-74.0999984741211" miny="40.75" maxx="-
74.0999984741211" maxy="40.75"/>
        <FIELDS NAME="New York" POPULATION="16472000"
#SHAPE#="[Geometry]" />
      </FEATURE>
      <FEATURECOUNT count="2" hasmore="false" />
      <ENVELOPE minx="-99.127571105957" miny="19.4270458221436" maxx="-
74.0999984741211" maxy="40.75"/>
```

```
    </FEATURES>
  </RESPONSE>
</ARCXML>
```

FEATURESERVERWORKSPACE

Used in: CONFIG MARKUP
Parent elements: WORKSPACES

<FEATURESERVERWORKSPACE

name ="*string*"
service ="*string*"
url ="*string*"
>

No Child Elements
</FEATURESERVERWORKSPACE >

Bold: Attribute or child element is required.

Description:

Specifies a workspace for an ArcIMS Feature Service.

Restrictions:

- Can only be used in viewer configuration files. It cannot be used in a map configuration file.

Notes:

None

Attribute Descriptions for FEATURESERVERWORKSPACE:

Attribute	Usage
name	Workspace name. Must be unique among all data sources.
service	ArcIMS service name.
url	URL points to location of ArcIMS servlet connector (servlet/com.esri.esrimap.Esrimap).

Examples for FEATURESERVERWORKSPACE:

Example 1: When in a viewer configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-178.215027" miny="18.924782" maxx="-66.969849"
maxy="71.406647" name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <FEATURESERVERWORKSPACE name="ifs_ws-0"
url="http://myserver.domain.com/servlet/com.esri.esrimap.Esrimap"
service="world" />
      </WORKSPACES>
      <LAYER type="featureclass" name="STATES" visible="true" id="0">
      <DATASET name="0" type="polygon" workspace="ifs_ws-0" />
      </LAYER>
      <LAYER type="featureclass" name="ROADS" visible="true" id="1">
      <DATASET name="3" type="line" workspace="ifs_ws-0" />
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

FIELD

Used in: CONFIG RESPONSE MARKUP
Servers: Image Query Feature Extract ArcMap
Parent elements: FCLASS FEATURE FIELDS SQVAR

<FIELD

*When parent element is **FIELDS:***
name ="***string***"
value ="*string*"

*When parent element is **SQVAR, FCLASS, FEATURE:***
name ="***string***"
type ="**-99 | -98 | -5 | -7 | 1 | 4 | 5 | 6 | 8 | 12 | 91**"
precision ="*integer*"
size ="*integer*"
>

*When parent element is **FEATURE:***
<FIELDVALUE... />

*When parent element is **SQVAR, FIELDS, FCLASS:***
No Child Elements

</FIELD >

Bold: Attribute or child element is required.

Description:

- When in a map configuration file or in MARKUP, FIELD is used to define the name of a field in the database.
- When in a response, it contains
  - The feature layer definition in LAYERINFO.
  - A geocoded address in GEOCODE.
  - The attribute information for a feature in FEATURE.

Restrictions:

None

Notes:

- In a FEATURE response, the field names #SHAPE# and #ID# are reserved for the shape column and unique feature ID, respectively. When #SHAPE# is used the *type* must be set to -98, and when #ID# is used, the type must be set to -99.
- In a FEATURE response, the value returned for dates is the number of milliseconds since 1970-01-01 00:00:00 GMT.
- For more details on using FIELDS in a FEATURE reponse, including additional examples, see Using GET_FEATURES and FEATURES.

Attribute Descriptions for FIELD:

*When parent element is **FIELDS:***

Attribute	Usage
name	Name of field in database.
value	Returns value from database field.

*When parent element is **SQVAR, FCLASS, FEATURE:***

Attribute	Usage
name	Name of field in database.
precision	Field precision as defined in database. Equal to number of decimal places.
size	Field size as defined in database. Must be 0 for shape fields.
type	Refers to the data type of the field as listed in the following table:

Type Value	Definition of Value
-99	Row_id fields.
-98	Shape fields.
-7	Boolean fields (true | false).
-5	Big integer fields. Values can range between -9223372036854775808 and 9223372036854775807 or 0 and 18446744073709551615.
1	CHAR fields. A value of 1 may be used in stored queries for strings in DBF files. After a stored query has been created and saved in ArcIMS Author, the value of 1 can be replaced with the value of 12. In stored queries using strings, 1 and 12 are interchangeable.
4	Integer fields. Values can range between -2147483648 and 2147483647 or 0 and 4294967295.
5	Small integer fields. Values can range between -

	32768 and 32767 or 0 and 65535.
6	Float fields. Values are single-precision floating-point numbers with allowable values between -3.402823466E+38 and -1.175494351E-38, 0, and between 1.175494351E-38 and 3.402823466E+38.
8	Double fields. Values are double-precision floating-point numbers with allowable values between -1.7976931348623157E+308 and -2.2250738585072014E-308, 0, and between 2.2250738585072014E-308 and 1.7976931348623157E+308.
12	String fields of any length.
91	Date fields.

Examples for FIELD:

Example 1: When in a stored query in CONFIG.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-141.0030059814453" miny="41.9133186340332"
maxx="-52.62028121948242" maxy="83.10832214355469"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to CANADA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="province" visible="true" id="0">
        <DATASET name="province" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltransparency="1.0"
fillcolor="127,127,127" />
        </SIMPLERENDERER>
        <EXTENSION type="StoredQuery">
          <STOREDQUERIES>
            <STOREDQUERY name="Provinces">
              <QUERY where="( NAME = &amp;apos;[%var%]&amp;apos; )"
subfields="#SHAPE# AREA CODE NAME POP1991 POP91_SQMI" />
              <SQVAR position="0" name="[%var%]">
                <FIELD name="NAME" precision="0" type="12" size="25"
```

```
/>
              </SQVAR>
            </STOREDQUERY>
          </STOREDQUERIES>
      </EXTENSION>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When in a SERVICEINFO response.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="Countries"
id="1">
        <FCLASS type="polygon">
          <FIELD name="AREA" type="8" size="12" precision="3" />
          <FIELD name="NAME" type="12" size="40" precision="0" />
          <FIELD name="ABBREVNAME" type="12" size="12" precision="0"
/>
          <FIELD name="FIPS_CODE" type="12" size="2" precision="0" />
          <FIELD name="WB_CNTRY" type="12" size="3" precision="0" />
          <FIELD name="HYPERLINK" type="12" size="60" precision="0" />
          <FIELD name="#SHAPE#" type="-98" size="0" precision="0" />
          <FIELD name="#ID#" type="-99" size="16" precision="0" />
        </FCLASS>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Example 3: When in a FEATURES response.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <FEATURES>
      <FEATURE>
        <FIELDS>
```

```
          <FIELD name="STATE_NAME" value="Washington" />
          <FIELD name="SUB_REGION" value="Pacific" />
        </FIELDS>
      </FEATURE>
      <FEATURECOUNT count="1" hasmore="false" />
    </FEATURES>
  </RESPONSE>
</ARCXML>
```

Example 4: When in a GEOCODE response.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<RESPONSE>
  <GEOCODE>
    <FEATURE featureid="1">
      <FIELD type="4" name="SCORE" size="5" precision="0">
        <FIELDVALUE valuestring="100" />
      </FIELD>
      <FIELD type="12" name="ADDRESSFOUND" size="21" precision="0">
        <FIELDVALUE valuestring="380 NEW YORK ST 92373" />
      </FIELD>
      <FIELD type="-98" name="SHAPEFIELD">
        <FIELDVALUE>
          <POINT x="-117.19496116" y="34.05777355" />
        </FIELDVALUE>
      </FIELD>
    </FEATURE>
    <GCCOUNT count="1" />
  </GEOCODE>
</RESPONSE>
</ARCXML>
```

Example 5: When in MARKUP.

```
<?xml version="1.0"?>

<MARKUP>
  <WORKSPACES>
    <FEATURESERVERWORKSPACE name="ifs_ws-20"
url="http://mymachine.domain.com/servlet/com.esri.esrimap.Esrimap"
service="world" />
  </WORKSPACES>
  <MARKUPLAYER layername="CNTRY94" workspace="ifs_ws-20">
    <MODIFIEDFEATURES>
      <FEATURE featureid="1000001">
      <ENVELOPE minx="79.7" miny="-59.0" maxx="113.9" maxy="-42.4" />
      <FIELD name="AREA" precision="3" size="12" type="8">
        <FIELDVALUE valuestring="10202" />
      </FIELD>
      <FIELD name="NAME" precision="0" size="40" type="12">
        <FIELDVALUE valuestring="Atlantis" />
      </FIELD>
```

```
      <FIELD name="#SHAPE#" precision="0" size="0" type="-98">
        <FIELDVALUE>
          <POLYGON>
            <RING>
            <POINT x="85.61944739721136" y="-42.43367913036056" />
            <POINT x="113.93068023991125" y="-46.33867676383642" />
            <POINT x="80.25007565118213" y="-55.12492143915705" />
            <POINT x="85.61944739721136" y="-42.43367913036056" />
            </RING>
          </POLYGON>
        </FIELDVALUE>
      </FIELD>
      </FEATURE>
    </MODIFIEDFEATURES>
  </MARKUPLAYER>
</MARKUP>
```

FIELDS

Used in: **RESPONSE**
Servers: Query Feature ArcMap
Parent elements: **FEATURE**

<FIELDS >
No Attributes

(m) <FIELD... />

</FIELDS >

(m): Child element can be used multiple times.

Description:

Provides the framework for a field description.

Restrictions:

None

Notes:

- If *outputmode="newxml"* in the GET_FEATURES request, then in the FEATURES response, FIELDS has no attributes but has FIELD as a child element:

```
<FIELDS>
    <FIELD name="CUST_ID" value="4" />
    <FIELD name="NAME" value="Customer 4" />
    <FIELD name="#SHAPE#" value="[Geometry]" />
    <FIELD name="#ID#" value="3" />
</FIELDS>
```

- If *outputmode="xml"* in a GET_FEATURES request, then in the FEATURES response, FIELDS has attributes describing all feature fields. Names of the attributes coincide with the names of the fields.

```
<FIELDS CUST_ID="4" NAME="Customer 4" #SHAPE#="[Geometry]"
#ID#="3" />
```

- The field names #SHAPE# and #ID# are reserved for the shape column and unique feature ID, respectively.
- The value returned for dates is the number of milliseconds since 1970-01-01 00:00:00 GMT.
- For more details on using FIELDS including additional examples, see Using GET_FEATURES and FEATURES.

Examples for FIELDS:

Example 1: When outputmode in GET_FEATURES is "xml".

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
<RESPONSE>
  <FEATURES>
    <FEATURE>
      <FIELDS AREA="1068927525.00000" PERIMETER="266301.07400"
NAME="Redlands" #SHAPE#="[Geometry]" #ID#="1" />
    </FEATURE>
    <FEATURECOUNT count="1" hasmore="false" />
  </FEATURES>
</RESPONSE>
</ARCXML>
```

Example 2: When outputmode in GET_FEATURES is "newxml".

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
<RESPONSE>
<FEATURES>
<FEATURE>
  <FIELDS>
   <FIELD name="AREA" value="1068927525.00000" />
   <FIELD name="PERIMETER" value="266301.07400" />
   <FIELD name="NAME" value="Redlands" />
   <FIELD name="#SHAPE#" value="[Geometry]" />
   <FIELD name="#ID#" value="1" />
  </FIELDS>
</FEATURE>
<FEATURECOUNT count="1" hasmore="false" />
</FEATURES>
</RESPONSE>
</ARCXML>
```

FIELDVALUE

Used in: **RESPONSE** **MARKUP**
Servers: Geocode
Parent elements: **FIELD**

<FIELDVALUE

valuestring ="*string*"
>

*When parent element is **FIELD in a GEOCODE response:***
<POINT... />

*When parent element is **FIELD in MARKUP:***
(m) <MULTIPOINT... /> *[Or]*
(m) <POLYGON... /> *[Or]*
(m) <POLYLINE... /> *[Or]*

</FIELDVALUE >

(m): Child element can be used multiple times.

Description:

Sets the value of a FIELD in a GEOCODE response and in MARKUP.

Restrictions:

None

Notes:

- In a GEOCODE response, the *valuestring* represents a range of 0 to 100 when FIELD *name* is "SCORE". The larger the number, the closer the address matches based on user input.

Attribute Descriptions for FIELDVALUE:

Attribute	Usage
valuestring	Returns field value from database field.

Examples for FIELDVALUE:

Example 1: When in a GEOCODE response.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<RESPONSE>
  <GEOCODE>
    <FEATURE featureid="1">
      <FIELD type="4" name="SCORE" size="5" precision="0">
        <FIELDVALUE valuestring="100" />
      </FIELD>
      <FIELD type="12" name="ADDRESSFOUND" size="21" precision="0">
        <FIELDVALUE valuestring="380 NEW YORK ST 92373" />
      </FIELD>
      <FIELD type="-98" name="SHAPEFIELD">
        <FIELDVALUE>
          <POINT x="-117.19496116" y="34.05777355" />
        </FIELDVALUE>
      </FIELD>
    </FEATURE>
    <GCCOUNT count="1" />
    </GEOCODE>
  </RESPONSE>
</ARCXML>
```

Example 2: When in MARKUP.

```
<?xml version="1.0"?>
<MARKUP>
  <WORKSPACES>
    <FEATURESERVERWORKSPACE name="ifs_ws-0"
url="http://mycomputer.domain.com/servlet/com.esri.esrimap.Esrimap"
service="world" />
  </WORKSPACES>
  <MARKUPLAYER layername="Countries" workspace="ifs_ws-0">
    <ADDEDFEATURES>
      <FEATURE featureid="1000000">
      <ENVELOPE minx="-35.92572247220596" miny="-48.809104594884616"
maxx="-5.291989044369359" maxy="-21.047283675907693" />
      <FIELD name="AREA" precision="3" size="12" type="8" />
      <FIELD name="NAME" precision="0" size="40" type="12" />
      <FIELD name="ABBREVNAME" precision="0" size="12" type="12" />
      <FIELD name="FIPS_CODE" precision="0" size="2" type="12" />
      <FIELD name="WB_CNTRY" precision="0" size="3" type="12" />
      <FIELD name="#SHAPE#" precision="0" size="0" type="-98">
        <FIELDVALUE>
          <POLYGON>
            <RING>
            <POINT x="-19.17289950385782" y="-21.047283675907693" />
            <POINT x="-35.92572247220596" y="-38.2787587290658" />
            <POINT x="-10.078509892468816" y="-48.809104594884616" />
            <POINT x="-5.291989044369359" y="-24.397848269577324" />
            <POINT x="-19.17289950385782" y="-21.047283675907693" />
            </RING>
```

```
          </POLYGON>
        </FIELDVALUE>
      </FIELD>
      </FEATURE>
    </ADDEDFEATURES>
  </MARKUPLAYER>
</MARKUP>
```

FILTERCOORDSYS

Used in: **CONFIG** **REQUEST**
Servers: Image Query Feature Extract ArcMap
Parent elements: **DATAFRAME** **PROPERTIES** **SPATIALQUERY**

<FILTERCOORDSYS

When using ArcMap Server:
id ="***integer***"
string ="***string***"

When using Image, Extract, Query, or Feature Server:
id ="***integer***"
string ="***string***"
datumtransformid ="*integer*"
datumtransformstring ="*string*"
>

No Child Elements
</FILTERCOORDSYS >

Bold: Attribute or child element is required.

Description:

The current coordinate system of filters in the requesting client.

Restrictions:

- Must use either *id* or *string*, but not both.
- For datum transformations either *datumtransformid* or *datumtransformstring* is used, but not both.
- In ArcMap Image Services, *datumtransformid* or *datumtransformstring* are not valid.

Notes:

- For a complete list of supported IDs and definition strings, see:
  - Projected Coordinate Systems Listing [Sorted by projection ID] [Sorted by name]
  - Geographic Coordinate Systems Listing [Sorted by projection ID] [Sorted by name]
  - Datum Transformation Listing [Sorted by projection ID] [Sorted by name]

- When using definition strings, the quotes in the string must be changed to " so the ArcIMS Spatial Server can interpret the string correctly. For example, the definition string for World Mollweide is:

  PROJCS["World_Mollweide",GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",0.0174532925199433295]],PROJECTION["Mollweide"],PARAMETER["False_Easting",0],PARAMETER["False_Northing",0],PARAMETER["Central_Meridian",0],UNIT["Meter",1]]

  Once the quotes have been changed, the string looks like this:

  PROJCS["World_Mollweide",GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",0.0174532925199433295]],PROJECTION["Mollweide"],PARAMETER["False_Easting",0],PARAMETER["False_Northing",0],PARAMETER["Central_Meridian",0],UNIT["Meter",1]]

- The attributes *datumtransformid* and *datumtransformstring* are used when datum transformation information needs to be included. Only datum transformations to and from WGS 1984 are supported.
  - When these attributes are used with COORDSYS and FILTERCOORDSYS, the datum transformation is from a non-WGS 1984 datum to WGS 1984. For example, Pulkovo_1942_To_WGS_1984 (datumtransformid="8157") transforms data from Pulkovo 1942 to WGS 1984.
  - When these attributes are used with FEATURECOORDSYS, the datum transformation is from WGS 1984 to a non-WGS 1984 datum. In the above example, the datum transformation is from WGS 1984 to Pulkovo 1942.

- If FEATURECOORDSYS is used, then FILTERCOORDSYS must also be included.
- If a service with FILTERCOORDSYS is used in the ArcIMS HTML Viewer, ArcIMS Java Standard or Custom Viewers, or ArcExplorer 4, FEATURECOORDSYS and FILTERCOORDSYS must use the same value for *id* or *string*.
- If a layer does not project, double check that a *.prj file or ArcSDE spatial reference table exists for the layer. If not, COORDSYS must be included with the layer.
- For more information on the projection elements, see Using Projection Elements.

Attribute Descriptions for FILTERCOORDSYS:

Attribute	Usage
datumtransformid	Datum transformation ID. Use either *datumtransformid* or *datumstransformstring*, but not both.
datumtransformstring	Datum transformation definition string. Use either *datumtransformid* or *datumstransformstring*, but not both.
id	Projected or geographic coordinate system ID. Use either *id* or *string*, but not both.
string	Projected or geographic coordinate system definition string. Use either *id* or *string*, but not both.

Examples for FILTERCOORDSYS:

Example 1: Using a coordinate system ID in a map configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <MAPUNITS units="meters" />
        <FEATURECOORDSYS id="54008" />
        <FILTERCOORDSYS id="4326" />
      </PROPERTIES>
      <WORKSPACES>
      <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="Cntry94" visible="true" id="0">
        <DATASET name="Cntry94" type="polygon" workspace="shp_ws-0" />
        <COORDSYS id="4326" />
        <SIMPLERENDERER>
           <SIMPLEPOLYGONSYMBOL filltransparency="1.0"
fillcolor="27,127,127" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: Using a coordinate system definition string.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-15066114" miny="-9182334" maxx="15311928"
maxy="8551044" />
        <MAPUNITS units="meters" />
        <FEATURECOORDSYS id="54008"/>
        <FILTERCOORDSYS
string="PROJCS[&quot;World_Mollweide&quot;,GEOGCS[&quot;GCS_WGS_1984&qu
ot;,DATUM[&quot;D_WGS_1984&quot;,SPHEROID[&quot;WGS_1984&quot;,6378137,
298.257223563]],PRIMEM[&quot;Greenwich&quot;,0],UNIT[&quot;Degree&quot;
,0.0174532925199433295]],PROJECTION[&quot;Mollweide&quot;],PARAMETER[&qu
ot;False_Easting&quot;,0],PARAMETER[&quot;False_Northing&quot;,0],PARAM
ETER[&quot;Central_Meridian&quot;,0],UNIT[&quot;Meter&quot;,1]]" />
      </PROPERTIES>
      <WORKSPACES>
      <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="Cntry94" visible="true" id="0">
        <DATASET name="Cntry94" type="polygon" workspace="shp_ws-0" />
        <COORDSYS id="4326" />
        <SIMPLERENDERER>
           <SIMPLEPOLYGONSYMBOL filltransparency="1.0"
fillcolor="27,127,127" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 3: When in a GET_IMAGE request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-145.0" miny="-30.0" maxx="-125.0" maxy="45.0"
/>
        <IMAGESIZE width="800" height="600" />
        <FEATURECOORDSYS id="54030"  />
        <FILTERCOORDSYS id="4326" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

FULLTEXT

Used in: REQUEST
Servers: Metadata (Browse)
Parent elements: SEARCH_METADATA

<FULLTEXT

word ="*string*"
>

No Child Elements
</FULLTEXT >

Bold: Attribute or child element is required.

Description:

Searches metadata document repository using one or more keywords.

Restrictions:

None

Notes:

None

Attribute Descriptions for FULLTEXT:

Attribute	Usage
word	List of one or more keywords.

Examples for FULLTEXT:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <SEARCH_METADATA operator="and">
        <ENVELOPE minx="-176.98" miny="18.93" maxx="-66.97"
maxy="71.35" spatialoperator="within" />
        <TAGVALUE
tag="metadata/dataqual/lineage/srcinfo/srccite/citeinfo/pubdate"
greaterthan="1990" />
        <FULLTEXT word="cities"/>
        <SEARCH_METADATA operator="or">
```

```
          <TAGTEXT tag="metadata/idinfo/citation/citeinfo/geoform"
word="data"/>
          <TAGTEXT tag="metadata/idinfo/citation/citeinfo/geoform"
word="digital"/>
        </SEARCH_METADATA>
      </SEARCH_METADATA>
    </GET_METADATA>
  </REQUEST>
</ARCXML>
```

GCCOUNT

Used in: RESPONSE
Servers: Geocode
Parent elements: GEOCODE

<GCCOUNT

count ***="integer"***
>

No Child Elements
</GCCOUNT >

Bold: Attribute or child element is required.

Description:

Contains number of candidates found during a geocode request.

Restrictions:

None

Notes:

- For more information on geocoding elements, see Summary of Geocoding Elements.

Attribute Descriptions for GCCOUNT:

Attribute	Usage
count	Number of candidates found.

Examples for GCCOUNT:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<RESPONSE>
  <GEOCODE>
    <FEATURE featureid="1">
      <FIELD type="4" name="SCORE" size="5" precision="0">
        <FIELDVALUE valuestring="100" />
      </FIELD>
      <FIELD type="12" name="ADDRESSFOUND" size="21" precision="0">
        <FIELDVALUE valuestring="380 NEW YORK ST 92373" />
      </FIELD>
```

```
      <FIELD type="-98" name="SHAPEFIELD">
        <FIELDVALUE>
          <POINT x="-117.19496116" y="34.05777355" />
        </FIELDVALUE>
      </FIELD>
    </FEATURE>
    <GCCOUNT count="1" />
  </GEOCODE>
</RESPONSE>
</ARCXML>
```

GCFIELD

Used in: CONFIG
Parent elements: GCSTYLE

<GCFIELD

id ="*string*"
name ="*string*"
>

No Child Elements
</GCFIELD >

Bold: Attribute or child element is required.

Description:

Defines the fields used for an address style.

Restrictions:

None

Notes:

- For more information on geocoding elements, see Summary of Geocoding Elements.

Attribute Descriptions for GCFIELD:

Attribute	Usage
id	The ID values for an address style. See the examples under GCSTYLE for the values needed for each style.
name	Name of field from database.

Examples for GCFIELD:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
```

```
    <MAP>
      <PROPERTIES>
       <ENVELOPE minx="-71.077092" miny="42.357962" maxx="-71.034511"
maxy="42.385263" name="Initial_Extent" />
       <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
       <SHAPEWORKSPACE name="shp_ws-64" directory="<path to data>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="Streets" visible="true" id="4">
       <DATASET name="bosstreets" type="line" workspace="shp_ws-64" />
       <SIMPLERENDERER>
         <SIMPLELINESYMBOL type="solid" width="2" color="255,0,0" />
       </SIMPLERENDERER>
       <EXTENSION type="Geocode">
         <GCSTYLE name="USAddressZ">
           <GCFIELD id="FromLeft" name="L_F_ADD" />
           <GCFIELD id="FromRight" name="R_F_ADD" />
           <GCFIELD id="ToLeft" name="L_T_ADD" />
           <GCFIELD id="ToRight" name="R_T_ADD" />
           <GCFIELD id="PreDir" name="PREFIX" />
           <GCFIELD id="PreType" name="PRE_TYPE" />
           <GCFIELD id="StreetName" name="NAME" />
           <GCFIELD id="StreetType" name="TYPE" />
           <GCFIELD id="SufDir" name="SUFFIX" />
           <GCFIELD id="LeftZone" name="ZIPL" />
           <GCFIELD id="RightZone" name="ZIPR" />
         </GCSTYLE>
       </EXTENSION>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

GCINPUT

Used in: RESPONSE
Servers: Geocode
Parent elements: GCSTYLE

<GCINPUT

description ="***string***"
id ="***string***"
label ="***string***"
width ="***integer***"
type ="text" **[text]**
>

No Child Elements
</GCINPUT >

Bold: Attribute or child element is required.

Description:

Returns the different ID values needed for an address style.

Restrictions:

None

Notes:

- GCINPUT is included in a SERVICEINFO response only if the GET_SERVICE_INFO request has been routed to the Geocode Server. This routing information is contained in the URL that is sent to the ArcIMS site such as in the following example (all one line):
  http://myComputer.domain.com/servlet/com.esri.esrimap.Esrimap?ClientVersion=3.1
  &ServiceName=myservice
  &CustomService=Geocode
  &Form=True&Encode=True
- All GCINPUT *id* attribute values are the same values used in GCTAG in a GET_GEOCODE request.
- For more information on geocoding elements, see Summary of Geocoding Elements.

Attribute Descriptions for GCINPUT:

Attribute	Usage
description	Description of field.
id	ID value for an address style.
label	Label for address dialog boxes.
type	Information used by the Java clients for type of output.
width	Width of input field used in Locate Address dialog box when using ArcIMS Java clients.

Examples for GCINPUT:

Example 1: When GET_SERVICE_INFO request is rerouted to the Geocode Server.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
      </ENVIRONMENT>
      <LAYERINFO name="Streets" id="4" >
        <EXTENSION type="geocode">
          <GCSTYLE name="USAddressZ" >
            <GCINPUT id="STREET" type="text" label="Street" width="10"
description="street number, street name and type" />
            <GCINPUT id="ZONE" type="text" label="Zone" width="5"
description="zone information" />
            <GCINPUT id="CROSSSTREET" type="text" label="Cross street"
width="10" description="cross street name and type" />
          </GCSTYLE>
        </EXTENSION>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

GCSTYLE

Used in: **CONFIG** **RESPONSE**
Servers: Geocode
Parent elements: **EXTENSION**

<GCSTYLE

name ="USAddressZ | USAddress | USSingleHouse | USSingleHouseZ | USSingleRange | USSingleRangeZ | Zip4 | Zip4Range | Zip5 | SingleField"
>

*When parent element is **EXTENSION in map configuration file:***
(m) <GCFIELD... />

*When parent element is **EXTENSION in SERVICEINFO response:***
(m) <GCINPUT... />

</GCSTYLE >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Identifies the geocoding style used for a layer.

Restrictions:

None

Notes:

- ArcIMS has 10 standard address styles:

GCSTYLE Address Style	Description
USAddressZ	U.S. streets with zone: This style considers street addresses with zone information such as postal ZIP Codes or city names. Street features are represented by two house number intervals, one for the left side of the street and the other for the right side of the street. The parity (odd/even) of each interval should agree. The beginning number of the intervals can either be lower or higher than the end number.
USAddress	U.S. streets: This style is used for address geocoding without zone information. Street features are represented

	by two house number intervals, one for the left side of the street and the other for the right side of the street.
USSingleRangeZ	U.S. single range with zone: This style considers street or parcel addresses with zone information such as postal ZIP Codes or city names. Features are represented by one house number interval. No parity (odd/even) of the house numbers is required.
USSingleRange	U.S. single range: This style is used for address geocoding without zone information. Features are represented by one house number interval. No parity (odd/even) of the house numbers is required.
USSingleHouseZ	U.S. single house with zone: This style considers parcel or point addresses with zone information such as postal ZIP Codes or city names. Features are represented by a single house number.
USSingleHouse	U.S. single house: This style is used for address geocoding without zone information. Features are represented by a single house number.
Zip4	ZIP+4: This style is used for geocoding ZIP+4 using two components (ZIP and ZIP4).
Zip4Range	ZIP+4 range: This style is used for geocoding ZIP+4 in a range using three components (ZIP, ZIP4Low, and ZIP4High).
Zip5	ZIP 5-digit: This style is used for geocoding five-digit ZIP Code addresses.
SingleField	Single field: This style is used for geocoding based on the value of a single field in the reference layer's attribute table. The style contains one component (Keyfield).

- ArcIMS supports the implementation of custom address styles. For information on customizing an address style for use in another country or for special cases in the United States, see Using Custom Address Styles.
- For more information on geocoding elements, see Summary of Geocoding Elements.

Attribute Descriptions for GCSTYLE:

Attribute	Usage
name	Geocoding style. See Notes section for a description of the different geocoding styles.

Examples for GCSTYLE:

Example 1: USAddressZ Style. Required elements are noted with <!--reqd-->.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
       <ENVELOPE minx="-71.077092" miny="42.357962" maxx="-71.034511"
maxy="42.385263" name="Initial_Extent" />
       <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
       <SHAPEWORKSPACE name="shp_ws-64" directory="<path to data>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="Streets" visible="true" id="4">
       <DATASET name="bosstreets" type="line" workspace="shp_ws-64" />
       <SIMPLERENDERER>
         <SIMPLELINESYMBOL type="solid" width="2" color="255,0,0" />
       </SIMPLERENDERER>
       <EXTENSION type="Geocode">
         <GCSTYLE name="USAddressZ">
<!--reqd--> <GCFIELD id="FromLeft" name="L_F_ADD" />
<!--reqd--> <GCFIELD id="FromRight" name="R_F_ADD" />
<!--reqd--> <GCFIELD id="ToLeft" name="L_T_ADD" />
<!--reqd--> <GCFIELD id="ToRight" name="R_T_ADD" />
            <GCFIELD id="PreDir" name="PREFIX" />
            <GCFIELD id="PreType" name="PRE_TYPE" />
<!--reqd--> <GCFIELD id="StreetName" name="NAME" />
            <GCFIELD id="StreetType" name="TYPE" />
            <GCFIELD id="SufDir" name="SUFFIX" />
<!--reqd--> <GCFIELD id="LeftZone" name="ZIPL" />
<!--reqd--> <GCFIELD id="RightZone" name="ZIPR" />
         </GCSTYLE>
       </EXTENSION>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: USAddress Style.

```
<EXTENSION type="Geocode">
          <GCSTYLE name="USAddress">
<!--reqd--> <GCFIELD id="FromLeft" name="L_F_ADD" />
<!--reqd--> <GCFIELD id="FromRight" name="R_F_ADD" />
<!--reqd--> <GCFIELD id="ToLeft" name="L_T_ADD" />
<!--reqd--> <GCFIELD id="ToRight" name="R_T_ADD" />
            <GCFIELD id="PreDir" name="PREFIX" />
```

```
            <GCFIELD id="PreType" name="PRE_TYPE" />
<!--reqd--> <GCFIELD id="StreetName" name="NAME" />
            <GCFIELD id="StreetType" name="TYPE" />
            <GCFIELD id="SufDir" name="SUFFIX" />
          </GCSTYLE>
</EXTENSION>
```

Example 3: USSingleHouseZ Style.

```
<EXTENSION type="Geocode">
   <GCSTYLE name="USSingleHouseZ">
<!--reqd--><GCFIELD id="HouseNum" name="HOUSE_NUM" />
     <GCFIELD id="PreDir" name="PRE_DIR" />
     <GCFIELD id="PreType" name="PRE_TYPE" />
<!--reqd--><GCFIELD id="StreetName" name="STREET_NAM" />
     <GCFIELD id="StreetType" name="STREET_TYP" />
     <GCFIELD id="SufDir" name="SUF_DIR" />
<!--reqd--><GCFIELD id="Zone" name="ZIP" />
   </GCSTYLE>
</EXTENSION>
```

Example 4: USSingleHouse Style.

```
<EXTENSION type="Geocode">
        <GCSTYLE name="USSingleHouse">
<!--reqd--> <GCFIELD id="HouseNum" name="ADDRESS" />
          <GCFIELD id="PreDir" name="PREFIX" />
          <GCFIELD id="PreType" name="PRE_TYPE" />
<!--reqd--> <GCFIELD id="StreetName" name="NAME" />
          <GCFIELD id="StreetType" name="TYPE" />
          <GCFIELD id="SufDir" name="SUFFIX" />
        </GCSTYLE>
</EXTENSION>
```

Example 5: USSingleRangeZ Style.

```
<EXTENSION type="Geocode">
        <GCSTYLE name="USSingleRangeZ">
<!--reqd--> <GCFIELD id="From" name="FROM_ADD" />
<!--reqd--> <GCFIELD id="To" name="TO_ADD" />
          <GCFIELD id="PreDir" name="PREFIX" />
          <GCFIELD id="PreType" name="PRE_TYPE" />
          <GCFIELD id="StreetName" name="NAME" />
<!--reqd--> <GCFIELD id="StreetType" name="TYPE" />
          <GCFIELD id="SufDir" name="SUFFIX" />
<!--reqd--> <GCFIELD id="Zone" name="CFCC" />
        </GCSTYLE>
</EXTENSION>
```

Example 6: USSingleRange Style.

```
<EXTENSION type="Geocode">
        <GCSTYLE name="USSingleRange">
<!--reqd--> <GCFIELD id="From" name="FROM_ADD" />
<!--reqd--> <GCFIELD id="To" name="TO_ADD" />
          <GCFIELD id="PreDir" name="PREFIX" />
          <GCFIELD id="PreType" name="PRE_TYPE" />
<!--reqd--> <GCFIELD id="StreetName" name="NAME" />
          <GCFIELD id="StreetType" name="TYPE" />
          <GCFIELD id="SufDir" name="SUFFIX" />
        </GCSTYLE>
</EXTENSION>
```

Example 7: Zip4 Style.

```
<EXTENSION type="Geocode">
        <GCSTYLE name="Zip4">
<!--reqd--> <GCFIELD id="ZIP" name="ZIP" />
<!--reqd--> <GCFIELD id="ZIP4" name="ZIP4" />
        </GCSTYLE>
</EXTENSION>
```

Example 8: Zip4Range Style.

```
<EXTENSION type="Geocode">
        <GCSTYLE name="Zip4Range">
<!--reqd--> <GCFIELD id="ZIP" name="ZIP" />
<!--reqd--> <GCFIELD id="Zip4Low" name="ZIPL" />
<!--reqd--> <GCFIELD id="Zip4High" name="ZIPH" />
        </GCSTYLE>
</EXTENSION>
```

Example 9: Zip5 Style.

```
<EXTENSION type="Geocode">
        <GCSTYLE name="Zip5">
<!--reqd--> <GCFIELD id="ZIP" name="ZIPL" />
        </GCSTYLE>
</EXTENSION>
```

Example 10: SingleField Style.

```
<EXTENSION type="Geocode">
        <GCSTYLE name="SingleField">
<!--reqd--> <GCFIELD id="KeyField" name="NAME" />
        </GCSTYLE>
</EXTENSION>
```

Example 11: When in a SERVICEINFO response.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96" />
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-71.077172" miny="42.355504" maxx="-71.034431"
maxy="42.387721" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="Streets"
id="0">
        <FCLASS type="line"></FCLASS>
        <EXTENSION type="Geocode" >
          <GCSTYLE name="USAddressZ" />
        </EXTENSION>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Example 12: When in a SERVICEINFO response that was routed to the Geocode Server.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
      </ENVIRONMENT>
      <LAYERINFO name="Streets" id="4" >
        <EXTENSION type="geocode">
          <GCSTYLE name="USAddressZ" >
            <GCINPUT id="STREET" type="text" label="Street" width="10"
description="street number, street name and type" />
            <GCINPUT id="ZONE" type="text" label="Zone" width="5"
description="zone information" />
            <GCINPUT id="CROSSSTREET" type="text" label="Cross street"
width="10" description="cross street name and type" />
          </GCSTYLE>
        </EXTENSION>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

GCTAG

Used in: REQUEST
Servers: Geocode
Parent elements: ADDRESS

```
<GCTAG

    id ="string"
    value ="string"
>

    No Child Elements
</GCTAG >
```

Bold: Attribute or child element is required.

Description:

Defines a single part of an address to be geocoded such as a street name or ZIP Code.

Restrictions:

None

Notes:

- The following table lists required and optional GCTAGs for each address style:

Address Styles	Required	Optional
USAddressZ, USSingleHouseZ, USSingleRangeZ	<GCTAG id="STREET" value="xxx" /> <GCTAG id="Zone" value="xxx" />	<GCTAG id="CrossStreet" value="xxx" />
USAddress, USSingleHouse, USSingleRange	<GCTAG id="STREET" value="xxx" />	<GCTAG id="CrossStreet" value="xxx" />
Zip4, Zip4Range, Zip5	<GCTAG id="Zip" value="xxx" />	
SingleField	<GCTAG id="Keyfield" value="xxx" />	

- All GCTAG *id* attribute values need to match GCINPUT values returned in SERVICEINFO.
- For more information on geocoding elements, see Summary of Geocoding Elements.

Attribute Descriptions for GCTAG:

Attribute	Usage
id	ID as defined in GCINPUT.
value	Values that user inputs for a geocoding request.

Examples for GCTAG:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_GEOCODE maxcandidates="25"  minscore="60">
      <LAYER id="streets" />
      <ADDRESS>
        <GCTAG id="STREET" value="380 New York st" />
        <GCTAG id="Zone" value="92373" />
        <GCTAG id="CrossStreet" value="" />
      </ADDRESS>
    </GET_GEOCODE>
  </REQUEST>
</ARCXML>
```

GEOCODE

Used in: RESPONSE
Servers: Geocode
Parent elements: RESPONSE

<GEOCODE >
No Attributes

<GCCOUNT... />
(m) <FEATURE... />

</GEOCODE >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Returns a list of address candidates and corresponding x,y coordinates.

Restrictions:

None

Notes:

- See GET_GEOCODE for request.
- For more information on geocoding elements, see Summary of Geocoding Elements.

Examples for GEOCODE:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<RESPONSE>
<GEOCODE>
  <FEATURE featureid="1">
    <FIELD type="4" name="SCORE" size="5" precision="0">
      <FIELDVALUE valuestring="100" />
    </FIELD>
    <FIELD type="12" name="ADDRESSFOUND" size="21" precision="0">
      <FIELDVALUE valuestring="380 NEW YORK ST 92373" />
    </FIELD>
    <FIELD type="-98" name="SHAPEFIELD">
      <FIELDVALUE>
        <POINT x="-117.19496116" y="34.05777355" />
      </FIELDVALUE>
    </FIELD>
  </FEATURE>
  <GCCOUNT count="1" />
```

```
</GEOCODE>
</RESPONSE>
</ARCXML>
```

GET_COLLECTION_INFO

Used in: REQUEST
Servers: Metadata (Browse)
Parent elements: GET_METADATA

```
<GET_COLLECTION_INFO

    collection ="words | tags"
    orderby ="alphabet | frequency | queries"   [alphabet]
    tag ="string"
    zcode ="string"
>

    No Child Elements
</GET_COLLECTION_INFO >
```

Bold: Attribute or child element is required.

Description:

Retrieves information about a metadata repository.

Restrictions:

None

Notes:

- See COLLECTION_INFO for response.
- When *collection="words"*, every unique word in the database is listed. However, the word must be significant. Words such as "the" or "is" are not included. This list can be long and unusable depending on the size of the database. To reduce the list of words, it is highly recommended to always use *tag*.
- When *collection="tags"*, all other attributes are ignored.
- When a metadata document is deleted, the words from that document remain in the collection. If a user runs a search on a word that is not in the collection, the word is added to the collection and marked as having been queried once but having no actual instances.

Attribute Descriptions for GET_COLLECTION_INFO:

Attribute	Usage
collection	A collection is a set of zero or more published documents. A document is the set of metadata describing a dataset. A document is either unpublished or published. Unpublished documents coexist with datasets on local file systems. Published documents are accessible over the Internet through catalog services.
orderby	When *collection* has the value of "words", specifices whether to order the word alphabetically, by the frequency of the word, or by the number of queries that have been issued against the word.
tag	Defines the location of an element inside an XML tree. For example, the element "title" might be inside an XML structure such as <idinfo> <citation> <citeinfo> **<title>...</title>** </citeinfo> </citation> </idinfo> To define "title", the string value for this attribute would be "idinfo/citation/citeinfo/**title**".
zcode	Numeric code defining semantics from the GEO Attribute Set.

Examples for GET_COLLECTION_INFO:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <GET_COLLECTION_INFO collection="words" orderby="frequency"
tag="metadata/dataqual/lineage/srcinfo/srccite/citeinfo/origin" />
    </GET_METADATA>
  </REQUEST>
</ARCXML>
```

GET_CONTENT_INFO

Used in: REQUEST
Servers: Metadata (Browse)
Parent elements: GET_METADATA

<GET_CONTENT_INFO >
No Attributes

No Child Elements
</GET_CONTENT_INFO >

Description:

Requests whether content is validated by clients.

Restrictions:

None

Notes:

- See CONTENT_INFO for response.
- The following items of content are central to the operations of searching and viewing search results in the Metadata Explorer. It is recommended that all published documents contain these items: Title, Publisher, Content type, Data theme, and Extent. These items are required if *validate* is "true" in METADATA_CONTENT.
- METADATA_CONTENT contains the attribute *index_words*. Information about this attribute is not available through GET_CONETNET_INFO. Only validataion information is returned.

Examples for GET_CONTENT_INFO:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <GET_CONTENT_INFO />
    </GET_METADATA>
  </REQUEST>
</ARCXML>
```

GET_EXTRACT

Used in: REQUEST
Servers: Extract
Parent elements: REQUEST

<GET_EXTRACT >
No Attributes

<ENVIRONMENT... />
(m) <LAYER... />
<PROPERTIES... />
<WORKSPACES... />

</GET_EXTRACT >

(m): Child element can be used multiple times.

Description:

Sends a request to the Extract Server. Layers are extracted into shapefile format and placed in a ZIP file.

Restrictions:

- Only vector data sets can be extracted. This includes shapefiles and ArcSDE layers. The layers can be extracted from an ArcIMS service or from dynamic vector layers in the GET_EXTRACT request. Image and acetate layers cannot be extracted.

Notes:

- See EXTRACT for response.
- In order for GET_EXTRACT to work on an ArcIMS service, the map configuration file must have at least one layer that includes an Extract EXTENSION. For example:

```
<EXTENSION type="extract" >
  <EXTRACTPARAMS clip="true" />
</EXTENSION>
```

- Extract requests must be routed to the Extract Server. This routing information is contained in the URL sent to the ArcIMS site such as in the following example (all one line):
http://myComputer.domain.com/servlet/com.esri.esrimap.Esrimap?ClientVersion=3.1
&ServiceName=myservice
&CustomService=Extract
&Form=True&Encode=True

- See Using GET_EXTRACT and EXTRACT for more information and additional examples.

Examples for GET_EXTRACT:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_EXTRACT>
     <PROPERTIES>
     <ENVELOPE minx="-130" miny="30" maxx="-90" maxy="60" />
        <LAYERLIST>
          <LAYERDEF id="Ocean" visible="true" />
          <LAYERDEF id="Countries" visible="true" />
          <LAYERDEF id="States" visible="true" />
          <LAYERDEF id="Provinces" visible="false" />
          <LAYERDEF id="Cities" visible="true" >
            <SPATIALQUERY>
              <SPATIALFILTER relation="area_intersection">
                <ENVELOPE minx="-120" miny="40" maxx="-100" maxy="60"
/>
              </SPATIALFILTER>
            </SPATIALQUERY>
         </LAYERDEF>
       </LAYERLIST>
     </PROPERTIES>
  </GET_EXTRACT>
</REQUEST>
</ARCXML>
```

GET_FEATURES

Used in: REQUEST
Servers: Query Feature ArcMap
Parent elements: REQUEST

<GET_FEATURES

attributes ="true | false" **[true]**
beginrecord ="*integer*" **[0]**
checkesc ="true | false" **[false]**
compact ="true | false" **[false]**
dataframe ="*string*"
envelope ="true | false" **[false]**
featurelimit ="*integer*" **[all features]**
geometry ="true | false" **[true]**
globalenvelope ="true | false" **[false]**
outputmode ="binary | xml | newxml" **[binary]**
skipfeatures ="true | false" **[false]**
>

<**LAYER... /**>
<**QUERY... /**> *[Or]*
<**SPATIALQUERY... /**> *[Or]*
<ENVIRONMENT... />

</GET_FEATURES >

Bold: Attribute or child element is required.

Description:

Requests features as a compressed binary stream from the Feature Server or requests attribute data in ArcXML format from the Query or ArcMap Servers.

Restrictions:

- Can only be used on one layer at a time. Layer must exist in the ArcIMS service.
- Either SPATIALQUERY or QUERY is required.
- *Skipfeatures* is valid only when *outputmode* is set to xml or newxml.
- Used only with Feature, Query, or ArcMap Server.
- The attribute *dataframe* is valid only with the ArcMap Server.

Notes:

- See FEATURES for response.
- A GET_FEATURES request to an Image Service gets routed to the Query Server. This routing information is contained in the URL sent to the ArcIMS site such as in the following example (all one line):
  http://myComputer.domain.com/servlet/com.esri.esrimap.Esrimap?ClientVersion

=3.1
&ServiceName=landuse
&CustomService=Query
&Form=True&Encode=True

Requests to an ArcMap Image Service do not need to be rerouted.

- The attributes *beginrecord* and *featurelimit* are used together to limit the number of features returned. *Featurelimit* is used to set the number of features returned. *Beginrecord* is used to start the retrieval at a specified record. For example, if 25 features should be returned, *featurelimit* is set to "25". For the first retrieval, *beginrecord* starts at "1" (the default). For the second retrieval, *beginrecord* should start at "26". To determine whether more than 25 features were extracted, the attribute *hasmore* in the FEATURES response will be set to "true" (more records) or "false" (no more records).
- *Outputmode* defines whether the data is streamed to the client or sent to the client in ArcXML format. If *outputmode* is binary, then the request goes to the Feature Server and the response is a compressed binary stream. This is the default and is valid only for Feature Services. To return data in ArcXML format, *outputmode* can have one of two values: newxml or xml.

  If *outputmode="xml"*, then all features are returned in a short ArcXML format using the FIELDS element:
  <FIELDS CUST_ID="4" NAME="Customer" #SHAPE#="[Geometry]" #ID#="3" />

  If *outputmode="newxml"* then all fields are returned in a longer ArcXML format in the FIELD element:
  ```
  <FIELDS>
    <FIELD name="CUST_ID" value="4" />
    <FIELD name="NAME" value="Customer" />
    <FIELD name="#SHAPE#" value="[Geometry]" />
    <FIELD name="#ID#" value="3" />
  </FIELDS>
  ```
- *Compact* can be used to switch the geometry used in the output ArcXML to a shortened format. When *compact* is true, the response uses the COORDS element:
  ```
  <MULTIPOINT>
    <COORDS>-122.406680 37.747422;-123,500555 37.820000</COORDS>
  </MULTIPOINT>
  ```

  When *compact* is false, the response uses the POINT element:
  ```
  <MULTIPOINT>
    <POINT x=-122.406680 y=37.747422 />
    <POINT x=-123,500555 y=37.820000 />
  </MULTIPOINT>
  ```
- For more details on using GET_FEATURES and FEATURES, including additional examples, see Using GET_FEATURES and FEATURES.

Attribute Descriptions for GET_FEATURES:

Attribute	Usage
attributes	Determines whether attribute data for selected features is returned in the response.
beginrecord	Index of first extracted record.
checkesc	Used to determine if the returned data should include escaped characters for ampersand, single quote, double quote, less than, and greater than. For example, if the value for SUB_REGION is "P&NW", then: • If checkesc="false", in the response the value of SUB_REGION is "P&NW". • If checkesc="true", the value of SUB_REGION is "P&NW".
compact	For ArcXML response, switches geometry to compact form. See Using GET_FEATURES and FEATURES for more details.
dataframe	Valid with ArcMap Server only. Dataframe image to use when requesting map. By default, the selected image is the default data frame in the ArcMap document. However, alternate data frames can be accessed using this attribute.
envelope	Used to request the bounding envelope of each returned feature.
featurelimit	Maximum number of returned features.
geometry	Requests feature coordinates.
globalenvelope	When set to true, the overall envelope for all the features returned is included in the response. Since the number of features returned depends on *beginrecord* and *featurelimit*, only the actual records extracted based on *featurelimit* are included in the overall envelope. Note that in order for the global envelope to be returned, *subfields* in the SPATIALQUERY must include either #SHAPE# or #ALL#.
outputmode	If *outputmode* is binary, data is compressed into a binary stream. This is valid only for Feature Services. If the value of "xml" is used, data is returned in ArcXML format. This format uses a shorthand format that is faster to parse in an HTML client. If "newxml" is used, the data is returned in a longer format. See Using GET_FEATURES and FEATURES for more details.
skipfeatures	When set to "true", only the number of features is returned without including any information on individual features. The number of features returned depends on *beginrecord* and *featurelimit*. To get the total number of features in a layer, a query can be specified with an empty *where* clause. Note that even when *skipfeatures* is set to "true", the query is executed and features are actually extracted before they are counted. This attribute is valid only when *outputmode* is "xml" or "newxml".

Examples for GET_FEATURES:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_FEATURES featurelimit="25" beginrecord="0" outputmode="xml"
geometry="false" envelope="true" compact="true">
    <LAYER id="4" />
    <SPATIALQUERY subfields="#ALL#" where="NAME  =  'Los Angeles'" >
    </SPATIALQUERY>
  </GET_FEATURES>
</REQUEST>
</ARCXML>
```

Example 2: When using projection elements.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES featurelimit="25" beginrecord="0" outputmode="xml"
geometry="true" envelope="true" compact="true">
      <LAYER id="4" />
      <SPATIALQUERY subfields="#ALL#" where="POPULATION > 10000000" >
        <FILTERCOORDSYS id="53030" />
        <FEATURECOORDSYS id="53030" />
        <SPATIALFILTER relation="area_intersection">
          <ENVELOPE minx="-11395772" miny="930558" maxx="-3878142"
maxy="6419621"/>
        </SPATIALFILTER>
      </SPATIALQUERY>
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

GET_GEOCODE

Used in: REQUEST
Servers: Geocode
Parent elements: REQUEST

```
<GET_GEOCODE

    maxcandidates ="integer"  [20]
    minscore ="0 – 100"  [60]
>

    <ADDRESS... />
    <LAYER... />
    <FEATURECOORDSYS... />

</GET_GEOCODE >
```

Bold: Attribute or child element is required.

Description:

Sends a request containing address information to the Geocode Server.

Restrictions:

- Used only with Geocode Server.

Notes:

- See GEOCODE for response.
- Geocode requests must be routed to the Geocode Server. This routing information is contained in the URL sent to the ArcIMS site such as in the following example (all one line):
  http://myComputer.domain.com/servlet/com.esri.esrimap.Esrimap?ClientVersion=3.1
  &ServiceName=myservice
  &CustomService=Geocode
  &Form=True&Encode=True
- Up to *maxcandidates* candidates with score greater than *minscore* is returned.
- FEATURECOORDSYS should be included only when the coordinate system of the ArcIMS service is different than the coordinate system in the client.
- For more information on geocoding elements, see Summary of Geocoding Elements.

Attribute Descriptions for GET_GEOCODE:

Attribute	Usage
maxcandidates	Maximum number of returned candidates.
minscore	Minimum score of returned candidates. If not included, all candidates with scores above 60 are returned. A candidate with a score of 100 means a perfect match, and 0 means no match.

Examples for GET_GEOCODE:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_GEOCODE maxcandidates="25"  minscore="60">
      <LAYER id="streets" />
      <ADDRESS>
        <GCTAG id="STREET" value="380 New York St" />
        <GCTAG id="Zone" value="92373" />
        <GCTAG id="CrossStreet" value="" />
      </ADDRESS>
    </GET_GEOCODE>
  </REQUEST>
</ARCXML>
```

GET_IMAGE

Used in: REQUEST
Servers: Image ArcMap
Parent elements: REQUEST

<GET_IMAGE

When using ArcMap Server:
autoresize ="true | false" **[false]**
dataframe ="*string*"

When using Image Server:
autoresize ="true | false" **[false]**
show ="layers"
>

<**PROPERTIES... /**>
<ENVIRONMENT... />
(m) <LAYER... />
<WORKSPACES... /> *[Image Server only]*

</GET_IMAGE >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Sends a request to the Image Server to generate a map image.

Restrictions:

- Valid workspace types include SHAPEWORKSPACE, IMAGEWORKSPACE, and SDEWORKSPACE and are valid with the Image Server only.

Notes:

- See IMAGE for response.
- For more details on using GET_IMAGE and IMAGE, including additional examples, see Using GET_IMAGE and IMAGE with Image Services or Using GET_IMAGE and IMAGE with ArcMap Image Services.

Attribute Descriptions for GET_IMAGE:

Attribute	Usage
dataframe	Valid with ArcMap Server only. Dataframe image to use when requesting map. By default, the selected image is the default data frame in the ArcMap document. However, alternate data frames can be accessed using this attribute.
autoresize	The maximum generated image size is based on the image memory limit set when an Image service is started. For example, an image memory limit of 1 MB allows a map no larger than 262,144 pixels (512 x 512) to be generated. If *autoresize* is set to "true", a requested map greater than the maximum pixel count will be reduced in size to within the maximum pixel count. If *autoresize* is set to "false", no image is generated and an error message is returned by the ArcIMS Spatial Server.
show	Use this attribute to return layer information in the response including layer ID, name, and number of features in the map.

Examples for GET_IMAGE:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
  <GET_IMAGE>
    <PROPERTIES>
      <LAYERLIST>
        <LAYERDEF id="1">
          <SIMPLERENDERER>
            <SIMPLEMARKERSYMBOL size="16" color="0,0,0" />
          </SIMPLERENDERER>
         <SPATIALQUERY>
            <SPATIALFILTER relation="area_intersection">
            <ENVELOPE maxy="60" maxx="60" miny="0" minx="0" />
         </SPATIALFILTER>
         </SPATIALQUERY>
      </LAYERDEF>
      </LAYERLIST>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
   </PROPERTIES>
  </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 2: Example using projections, queries, selected features, and acetate layers. The IMAGE response includes layer information since the attribute "show" is used.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE show="layers">
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <IMAGESIZE width="800" height="600" />
        <FEATURECOORDSYS id="54030"  />
        <FILTERCOORDSYS id="4326" />
        <LAYERLIST>
          <LAYERDEF id="0" visible="true" />
          <LAYERDEF id="1" visible="true" />
          <LAYERDEF id="2" visible="false" />
          <LAYERDEF id="3" visible="false" />
          <LAYERDEF id="4" visible="true" >
            <SPATIALQUERY where="POPULATION &gt; 2000000" >
              <SPATIALFILTER relation="area_intersection">
                <ENVELOPE minx="-14.0" miny="35.0" maxx="33.0"
maxy="64.0" />
              </SPATIALFILTER>
            </SPATIALQUERY>
            <SIMPLERENDERER>
              <SIMPLEMARKERSYMBOL type="star" color="0,155,0"
width="12.0" />
            </SIMPLERENDERER>
          </LAYERDEF>
        </LAYERLIST>
      </PROPERTIES>

      <LAYER type="featureclass" name="new_CNTRY94" id="333" >
        <DATASET fromlayer="1" />
        <SPATIALQUERY where="NAME=&apos;Brazil&apos;" />
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0"
/>
          </SIMPLERENDERER>
      </LAYER>

      <LAYER type="acetate" name="WorldText" id="444" >
        <OBJECT units="pixel">
          <TEXT size="32" coords="100 44" label="The World">
            <TEXTMARKERSYMBOL fontstyle="bold" fontsize="32"
font="Arial" fontcolor="0,0,0" glowing="255,255,0" />
          </TEXT>
        </OBJECT>
      </LAYER>

      <LAYER type="acetate" name="WorldBox" id="3333">
        <OBJECT units="pixel">
          <LINE coords="10 40;400 40;400 80;10 80;10 40">
          <SIMPLELINESYMBOL color="0,0,0" />
          </LINE>
```

```
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 3: When using the attribute dataframe with GET_IMAGE.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE dataframe="States">
      <PROPERTIES>
        <ENVELOPE minx="-94" miny="32" maxx="-73" maxy="46" />
        <IMAGESIZE width="500" height="400" />
        <LAYERLIST>
          <LAYERDEF id="2" visible="true" /> <!--States-->
          <LAYERDEF id="1" visible="false" /> <!--Roads-->
          <LAYERDEF id="0" visible="false" /> <!--Cities-->
        </LAYERLIST>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

GET_LAYOUT

Used in: **REQUEST**
Servers: ArcMap
Parent elements: **REQUEST**

<GET_LAYOUT

autoresize ="true | false" **[false]**
>

(m) <DATAFRAME... />
<PROPERTIES... />

</GET_LAYOUT >

(m): Child element can be used multiple times.

Description:

Requests an ArcMap layout.

Restrictions:

None

Notes:

- See LAYOUT for response.
- When ENVELOPE is used in PROPERTIES, the coordinates are in the page coordinates of the layout. When used in DATAFRAME, the coordinates are in map units.
- For more details on using GET_LAYOUT and LAYOUT, including additional examples, see Using GET_LAYOUT and LAYOUT with ArcMap Image Services.

Attribute Descriptions for GET_LAYOUT:

Attribute	Usage
autoresize	The maximum generated layout size is based on the image memory limit set when an ArcMap service is started. For example, an image memory limit of 1 MB allows a map no larger than 262,144 pixels (512 x 512) to be generated. If *autoresize* is set to "true", a requested layout greater than the maximum pixel count is reduced in size to within the maximum pixel count. If *autoresize* is set to "false", no image is generated and an error message is returned by the ArcIMS Spatial Server. Output formats affected by *autoresize* are PNG8, PNG24, GIF, and JPG.

Examples for GET_LAYOUT:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_LAYOUT>
      <PROPERTIES>
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
        <FILTERCOORDSYS id="54030" />
        <FEATURECOORDSYS id="54030" />
        <IMAGESIZE width="800" height="600" />
        <OUTPUT type="JPG" />
      </PROPERTIES>
      <DATAFRAME id="Layers" >
        <FILTERCOORDSYS id="4326" />
        <FEATURECOORDSYS id="4326" />
        <ENVELOPE minx="-121" miny="36" maxx="-112" maxy="44" />
      </DATAFRAME>
    </GET_LAYOUT>
  </REQUEST>
</ARCXML>
```

GET_METADATA

Used in: REQUEST
Servers: Metadata (Browse)
Parent elements: REQUEST

<GET_METADATA >
No Attributes

<ADD_RELEVANCE_FEEDBACK... /> *[Or]*
<GET_COLLECTION_INFO... /> *[Or]*
<GET_CONTENT_INFO... /> *[Or]*
<GET_METADATA_DOCUMENT... /> *[Or]*
<GET_ROOT_DATASET... /> *[Or]*
<GET_USER... /> *[Or]*
<SEARCH_METADATA... /> *[Or]*

</GET_METADATA >

Bold: Attribute or child element is required.

Description:

Parent element used to send requests to the Metadata Server.

Restrictions:

- Only one child element can be used per request.

Notes:

- See METADATA for response.

Examples for GET_METADATA:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <GET_COLLECTION_INFO collection="words" orderby="frequency"
tag="metadata/dataqual/lineage/srcinfo/srccite/citeinfo/origin" />
    </GET_METADATA>
  </REQUEST>
</ARCXML>
```

GET_METADATA_DOCUMENT

Used in: REQUEST
Servers: Metadata (Browse)
Parent elements: GET_METADATA

```
<GET_METADATA_DOCUMENT

    docid ="string"
>

    No Child Elements
</GET_METADATA_DOCUMENT >
```

Bold: Attribute or child element is required.

Description:

Used by a client to retrieve a metadata document from the server.

Restrictions:

None

Notes:

- See METADATA_DATASET for response.

Attribute Descriptions for GET_METADATA_DOCUMENT:

Attribute	Usage
docid	String used to uniquely identify a document. The client used to publish the metadata is responsible for creating the document ID. This ID is automatically assigned when using ArcCatalog. If another client is used, GET_UUID can be used to request a valid ID. The format for an ID is the following: {HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH} where H is a hexadecimal digit (0-9,a-f,A-F). The ID is limited to 38 characters.

Examples for GET_METADATA_DOCUMENT:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <GET_METADATA_DOCUMENT docid="{1D4C8198-9409-11D5-99E8-
000086460FA0}" />
    </GET_METADATA>
  </REQUEST>
</ARCXML>
```

GET_RASTER_INFO

Used in: **REQUEST**
Servers: Image ArcMap
Parent elements: **REQUEST**

<GET_RASTER_INFO

When using ArcMap Server:
layerid ="***string***"
x ="***double***"
y ="***double***"
dataframe ="*string*"

When using Image Server:
layerid ="***string***"
x ="***double***"
y ="***double***"
>

<COORDSYS... />

</GET_RASTER_INFO >

Bold: Attribute or child element is required.

Description:

Returns the pixel value of an image at a given x,y coordinate location for the specified layer.

Restrictions:

- Valid with Image and ArcMap Server only.

Notes:

- See RASTER_INFO for response.
- If COORDSYS is not included as a child element, it is assumed that the coordinates used for GET_RASTER_INFO *x* and *y* are in the same coordinate system as the raster image. If COORDSYS is used in the request, the attributes are in the coordinate system defined by COORDSYS.

Attribute Descriptions for GET_RASTER_INFO:

Attribute	Usage
dataframe	Valid with ArcMap Server only. Dataframe image to use when requesting map. By default, the selected image is the default data frame in the ArcMap document. However, alternate data frames can be accessed using this attribute.
layerid	ID of the layer to query.
x	X-coordinate of selected point.
y	Y-coordinate of selected point.

Examples for GET_RASTER_INFO:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_RASTER_INFO x="136.333846" y="60.075385" layerid="0" >
      <COORDSYS id="4326" />
    </GET_RASTER_INFO>
  </REQUEST>
</ARCXML>
```

GET_ROOT_DATASET

Used in: REQUEST
Servers: Metadata (Browse)
Parent elements: GET_METADATA

<GET_ROOT_DATASET >
No Attributes

No Child Elements
</GET_ROOT_DATASET >

Description:

Retrieves the root document.

Restrictions:

None

Notes:

- See METADATA_DATASET for response.

Examples for GET_ROOT_DATASET:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <GET_ROOT_DATASET />
    </GET_METADATA>
  </REQUEST>
</ARCXML>
```

GET_SERVICE_INFO

Used in: REQUEST
Servers: Image Query Feature Extract Geocode ArcMap
Parent elements: REQUEST

<GET_SERVICE_INFO

*When parent element is **REQUEST for an ArcMap Image Service:***
dataframe ="#ALL# | frame by name"
dpi ="1 - NNN"
envelope ="true | false" **[true]**
fields ="true | false" **[true]**
toc ="true | false" **[false]**
toctype ="jpg | png8 | png24 | bmp" **[jpg]**

*When parent element is **REQUEST for an Image or Feature Service:***
dpi ="1 - NNN"
envelope ="true | false" **[true]**
extensions ="true | false" **[true]**
fields ="true | false" **[true]**
renderer ="true | false" **[true]**
>

No Child Elements
</GET_SERVICE_INFO >

Description:

Requests information about each layer in an ArcIMS service. With Image and Feature Services, the 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, dataframes, and the table of contents.

Restrictions:

- Not valid with Metadata Server.

Notes:

- See SERVICEINFO for response.
- A GET_SERVICE_INFO request sent to the Image or Feature Server returns limited information about Geocode extensions in an ArcIMS service. In order to retrieve all the information about a Geocode extension, the GET_SERVICE_INFO request must be routed to the Geocode Server. This routing information is contained in the URL sent to the ArcIMS site such as in the following example (all one line):

http://myComputer.domain.com/servlet/com.esri.esrimap.Esrimap?ClientVersion=3.1
&ServiceName=myservice
&**CustomService=Geocode**
&Form=True&Encode=True

- For more details on using GET_SERVICE_INFO and SERVICEINFO including additional examples, see Using GET_SERVICE_INFO and SERVICEINFO with Image and Feature Services and Using GET_SERVICE_INFO and SERVICEINFO with ArcMap Image Services.

Attribute Descriptions for GET_SERVICE_INFO:

*When parent element is **REQUEST for an ArcMap Image Service:***

Attribute	Usage
dataframe	Retrieves dataframe information for an ArcMap Image Service using layouts. Use the value "#ALL#" to retrieve information for all frames. Information for a single dataframe can be retrieved by specifying the frame's name. Multiple data frame names can be used and should be separated by a semicolon.
dpi	Dots per inch (dpi). Used for calculating the correct scale thresholds for layers in the service. The dpi must be calculated by the client. Otherwise, a value of "96" is assumed.
envelope	Toggle to get information on the envelope for layers in a service.
fields	Toggle to get information on available fields for each featureclass layer in the service.
toc	Toggle to get information on the table of contents (TOC) used to generate an ArcMap Image Service legend for the ArcIMS Java Viewers and ArcMap. Not used with ArcIMS HTML Viewers. Instead a legend is retrieved using LEGEND in a GET_IMAGE request.
toctype	Determines image format for an ArcMap Image Service legend.

*When parent element is **REQUEST for an Image or Feature Service:***

Attribute	Usage
dpi	Dots per inch (dpi). Used for calculating the correct scale thresholds for scale dependent elements such as SCALEDEPENDENTRENDERER, LAYER, and OBJECT. The *dpi* value used in request overrides the value used in a service.
envelope	Toggle to get information on the envelope for featureclass layers in a service. Featureclass layers include shapefile and ArcSDE layers.
extensions	Toggle to get information on any service extensions in featureclass

	layers.
fields	Toggle to get information on available fields for each featureclass layer in a service.
renderer	Toggle to get renderer information for featureclass layers in a service.

Examples for GET_SERVICE_INFO:

Example 1: When using Image or Feature Services.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_SERVICE_INFO fields="false" envelope="false"
renderer="false" extensions="true" />
  </REQUEST>
</ARCXML>
```

Example 2: When using ArcMap Image Services.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_SERVICE_INFO fields="false" envelope="false"
dataframe="#ALL# toc="true" toctype="jpg" />
  </REQUEST>
</ARCXML>
```

GET_USER

Used in: REQUEST
Servers: Metadata (Browse)
Parent elements: GET_METADATA

<GET_USER

username ***="string"***
>

No Child Elements
</GET_USER >

Bold: Attribute or child element is required.

Description:

Requests user information by user name.

Restrictions:

- Not available from ArcCatalog.

Notes:

- See USER for response.

Attribute Descriptions for GET_USER:

Attribute	Usage
username	Name of user.

Examples for GET_USER:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <GET_USER username="aUser" />
    </GET_METADATA>
  </REQUEST>
</ARCXML>
```

GET_UUID

Used in: REQUEST
Servers: Metadata (Publish)
Parent elements: PUBLISH_METADATA

<GET_UUID

count ="*integer*" **[1]**
>

No Child Elements
</GET_UUID >

Description:

Requests a unique ID that can be used as a document ID.

Restrictions:

None

Notes:

- See UUID for response.

Attribute Descriptions for GET_UUID:

Attribute	Usage
count	Determines the number of unique IDs to generate.

Examples for GET_UUID:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <PUBLISH_METADATA>
      <GET_UUID count="5" />
    </PUBLISH_METADATA>
  </REQUEST>
</ARCXML>
```

GETCLIENTSERVICES

Used in: Application Server REQUEST
Parent elements: None

<GETCLIENTSERVICES >
No Attributes

No Child Elements
</GETCLIENTSERVICES >

Description:

Gets a listing of ArcIMS services available on a host site.

Restrictions:

None

Notes:

- See SERVICES for response.

Examples for GETCLIENTSERVICES:

Example 1:

```
<GETCLIENTSERVICES/>
```

GRADIENTFILLSYMBOL

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Feature ArcMap
Parent elements: **EXACT** **OBJECT** **OTHER** **POLYGON** **RANGE** **SIMPLERENDERER**

<GRADIENTFILLSYMBOL

When using ArcMap Server:
finishcolor ="0,0,0 - 255,255,255" **[0,255,0]**
startcolor ="0,0,0 - 255,255,255" **[255,0,0]**
type ="bdiagonal | fdiagonal | horizontal | vertical" **[bdiagonal]**

When using Image or Feature Server:
antialiasing ="true | false" **[false]**
finishcolor ="0,0,0 - 255,255,255" **[0,255,0]**
overlap ="true | false" **[true]**
startcolor ="0,0,0 - 255,255,255" **[255,0,0]**
transparency ="0.0 - 1.0" **[1.0]**
type ="bdiagonal | fdiagonal | horizontal | vertical" **[bdiagonal]**
>

No Child Elements
</GRADIENTFILLSYMBOL >

Description:

Fills polygons with a gradual gradient based on two colors.

Restrictions:

- *Overlap* is valid only for Image Services. It is ignored when using Feature Services.
- In ArcMap Image Services, symbol is valid only in acetate layers.

Notes:

- This element is not available from the ArcIMS Author interface.

Attribute Descriptions for GRADIENTFILLSYMBOL:

Attribute	Usage
antialiasing	Used to make edges of labels and symbols smoother. When set to "true", antialiasing is active. Note that the time to generate a map may significantly increase when antialiasing is turned on.
finishcolor	End color using RGB values.
overlap	Determines if labels can overlap this symbol. When "true", labels can overlap. When "false", labels will not overlap the symbol. If labels are not drawing as expected, check if *overlap* is set to "false" for this symbol or any other symbol in the ArcIMS service.
startcolor	Start color using RGB values.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.
type	Symbol type.

Examples for GRADIENTFILLSYMBOL:

Example 1: When in CONFIG or REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
         <ENVELOPE minx="-124.594842" miny="24.955227" maxx="-
67.672764" maxy="49.596039" name="Initial_Extent" />
          <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="STATES" visible="true" id="1">
        <DATASET name="STATES" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <GRADIENTFILLSYMBOL transparency="1.0" type="vertical"
startcolor="0,255,0" finishcolor="0,0,255" overlap="true" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

GROUPRENDERER

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Feature
Parent elements: **GROUPRENDERER** **LAYER** **LAYERDEF** **LAYERINFO** **SCALEDEPENDENTRENDERER**

<GROUPRENDERER >
No Attributes

(m) <GROUPRENDERER... />
(m) <SCALEDEPENDENTRENDERER... />
(m) <SIMPLELABELRENDERER... />
(m) <SIMPLERENDERER... />
(m) <VALUEMAPLABELRENDERER... />
(m) <VALUEMAPRENDERER... />

</GROUPRENDERER >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Groups two or more renderers together. Common uses are to group feature rendering and labeling, to use multiple scale-dependent renderers on the same layer, and to create complex symbology.

Restrictions:

- At least one child element is required. Any combination and number of child elements can be used.
- Not valid with ArcMap Server.

Notes:

- For more information on using renderers, see Using ArcXML Renderers.

Examples for GROUPRENDERER:

Example 1: Grouping a symbol and a label.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CITIES" visible="true" id="2">
        <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <SIMPLERENDERER>
            <SIMPLEMARKERSYMBOL color="227,127,227" width="6" />
          </SIMPLERENDERER>
          <SIMPLELABELRENDERER field="CITY_NAME"
labelpriorities="0,0,1,0,0,0,0,0">
            <TEXTSYMBOL antialiasing="true" font="Arial"
fontstyle="bold" fontsize="12" printmode="alllower"/>
          </SIMPLELABELRENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

HASHLINESYMBOL

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Feature ArcMap
Parent elements: **EXACT** **LINE** **OBJECT** **OTHER** **POLYGON** **RANGE** **SIMPLERENDERER**

<HASHLINESYMBOL

When using ArcMap Server:
color ="0,0,0 - 255,255,255" **[0,0,0]**
interval ="0 - NNN" **[8]**
linethickness ="1 - NNN" **[1]**
tickthickness ="1 - NNN" **[1]**
type ="foreground | background" **[foreground]**
width ="1 - NNN" **[6]**

When using Image or Feature Server:
antialiasing ="true | false" **[false]**
color ="0,0,0 - 255,255,255" **[0,0,0]**
interval ="0 - NNN" **[8]**
linethickness ="1 - NNN" **[1]**
overlap ="true | false" **[true]**
tickthickness ="1 - NNN" **[1]**
transparency ="0.0 - 1.0" **[1.0]**
type ="foreground | background" **[foreground]**
width ="1 - NNN" **[6]**
>

No Child Elements
</HASHLINESYMBOL >

Description:

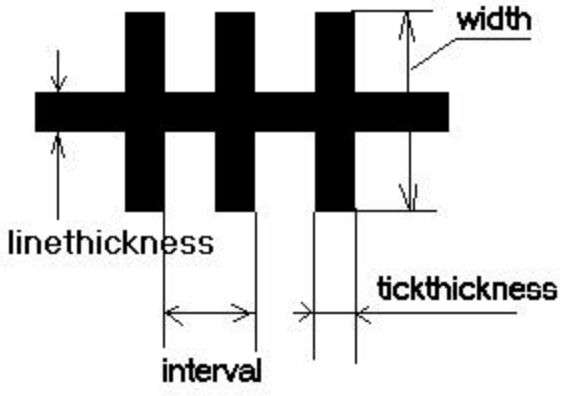
Line symbol for drawing railroad symbols.

Restrictions:

- *Overlap* is valid only for Image Services. It is ignored when using Feature Services.
- Not valid with ArcMap Server.

Notes:

- The following figure shows the different components that make up HASHLINESYMBOL:



- HASHLINESYMBOL uses a smoothing algorithm on the line to get a better hash effect. If a different line symbol is used, the smoothing algorithm is not applied, and a line will not overlay exactly with a hashline (railroad). In order to make a line overlay hashline, use HASHLINESYMBOL for the line and set the attribute *type* to background.

Attribute Descriptions for HASHLINESYMBOL:

Attribute	Usage
antialiasing	Used to make edges of labels and symbols smoother. When set to "true", antialiasing is active. Note that the time to generate a map may significantly increase when antialiasing is turned on.
color	Symbol color using RGB values.
interval	Distance between railroad crosshatches in pixels.
linethickness	Line thickness in pixels.
overlap	Determines if labels can overlap this symbol. When "true", labels can overlap. When "false", labels will not overlap the symbol. If labels are not drawing as expected, check if *overlap* is set to "false" for this symbol or any other symbol in the ArcIMS service.
tickthickness	Tick thickness in pixels.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.
type	If *type* is "foreground", the symbol draws as a railroad with the crosshash. If *type* is "background", the symbol draws as a simple line without the crosshash.
width	Width of the crosshash segments in pixels.

Examples for HASHLINESYMBOL:

Example 1: When in CONFIG or REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
         <ENVELOPE minx="-124.594842" miny="24.955227" maxx="-
67.672764" maxy="49.596039" name="Initial_Extent" />
          <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="ROADS" visible="true" id="1">
        <DATASET name="ROADS" type="line" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <HASHLINESYMBOL color="127,227,27" linethickness="8"
tickthickness="8" transparency="0.5" interval="16" width="16"
type="foreground" antialiasing="false" overlap="true" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

HOLE

Used in: **CONFIG** **REQUEST** **RESPONSE** **MARKUP**
Servers: Image Query Feature Extract ArcMap
Parent elements: **RING**

<HOLE >
No Attributes

<COORDS... /> *[Or]*
(m) <POINT... /> *[Or]*

</HOLE >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Provides the x,y coordinate locations for holes inside a polygon feature.

Restrictions:

- Either COORDS or POINT is required.

Notes:

- For the different options of representing geometry in an acetate layer, see OBJECT.

Examples for HOLE:

Example 1: When in SPATIALFILTER in CONFIG or REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_IMAGE>
    <PROPERTIES>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      <IMAGESIZE width="643" height="502" />
    </PROPERTIES>
    <LAYER type="featureclass" name="select layer" visible="true"
id="selected">
      <DATASET fromlayer="countries" />
      <SPATIALQUERY>
        <SPATIALFILTER relation="area_intersection">
          <POLYGON>
            <RING>
              <POINT x="-133.15605550814075" y="78.07185101549165" />
              <POINT x="-131.09942196116728" y="74.70645066589869" />
              <POINT x="-128.1079549837513" y="76.38915084069517" />
```

```
            <POINT x="-128.1079549837513" y="76.38915084069517" />
            <POINT x="-133.15605550814075" y="78.07185101549165" />
            <HOLE>
              <POINT x="-135.15605550814075" y="75.07185101549165" />
              <POINT x="-137.09942196116728" y="72.70645066589869" />
              <POINT x="-130.1079549837513" y="79.38915084069517"/>
            </HOLE>
          </RING>
        </POLYGON>
      </SPATIALFILTER>
    </SPATIALQUERY>
    <SIMPLERENDERER>
      <SIMPLEPOLYGONSYMBOL fillcolor="0,0,0" filltype="cross" />
    </SIMPLERENDERER>
  </LAYER>
 </GET_IMAGE>
</REQUEST>
</ARCXML>
```

Example 2: When using HOLE in an acetate layer.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <IMAGESIZE width="800" height="600" />
      </PROPERTIES>
      <LAYER type="acetate" name="acetate1" id="acetate1">
        <OBJECT units="database">
          <SIMPLEPOLYGONSYMBOL fillcolor="0,255,0" />
          <POLYGON>
            <RING>
              <POINT x="83.15605550814075" y="38.07185101549165" />
              <POINT x="111.09942196116728" y="-4.70645066589869" />
              <POINT x="155.1079549837513" y="-10.38915084069517" />
              <POINT x="139.1079549837513" y="66.38915084069517" />
              <POINT x="83.15605550814075" y="38.07185101549165" />
              <HOLE>
                <POINT x="100.15605550814075" y="20.07185101549165" />
                <POINT x="103.09942196116728" y="30.70645066589869" />
                <POINT x="106.1079549837513" y="30.38915084069517" />
                <POINT x="100.15605550814075" y="20.07185101549165" />
              </HOLE>
            </RING>
          </POLYGON>
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

IMAGE

Used in: RESPONSE
Servers: Image ArcMap
Parent elements: RESPONSE

<IMAGE >
No Attributes

<**ENVELOPE... /**>
<LAYERS... />
<LEGEND... />
<OUTPUT... />

</IMAGE >

Bold: Attribute or child element is required.

Description:

The main element for a response from an Image or ArcMap Image Service. Returns the location of a map image or legend image. Also can return information such as layer names, IDs, and number of features.

Restrictions:

None

Notes:

- See GET_IMAGE for request.
- OUTPUT is always returned unless DRAW in a GET_IMAGE request has the attribute *map* set to "false".
- LEGEND is returned with the legend's location if LEGEND is used in PROPERTIES of a GET_IMAGE request. LEGEND is always returned if it is included in the PROPERTIES section of a map configuration file.
- For more details on using GET_IMAGE and IMAGE including additional examples, see Using GET_IMAGE and IMAGE with Image Services or Using GET_IMAGE and IMAGE with ArcMap Image Services.

Examples for IMAGE:

Example 1:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-118.19793324" miny="34.03441917" maxx="-
118.12940130" maxy="34.08010713" />
      <OUTPUT file="c:\output\world_MYMACHINE2052765.gif"
url="http://mymachine.domain.com/output/world_MYMACHINE2052765.gif" />
      <LEGEND file="c:\output\world_MYMACHINE2052766.gif"
url="http://mymachine.domain.com/output/world_MYMACHINE2052766.gif" />
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

Example 2: Response showing details on layers in the requested MapService.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-180" miny="-135" maxx="180" maxy="135" />
      <LAYERS>
        <LAYER name="CNTRY94" id="0" featurecount="165" />
        <LAYER name="STATES" id="1" featurecount="51" />
        <LAYER name="acetate" id="2" featurecount="1" />
      </LAYERS>
      <OUTPUT file="c:\arcims\output\world_MYMACHINE3633699.jpg"
url="http://mymachine.domain.com/output/world_MYMACHINE3633699.jpg"
/>
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

IMAGEGENERALIZATION

Used in: **CONFIG** **RESPONSE**
Servers: Image
Parent elements: **PROPERTIES**

```
<IMAGEGENERALIZATION

    mode ="true | false"   [true]
>

    No Child Elements
</IMAGEGENERALIZATION >
```

Description:

Turns off Image Server generalization so that small or narrow features do not disappear.

Restrictions:

- Not valid in requests.

Notes:

- Generalization is used by the Image Server to speed up the processing of features, and by default, features are generalized. However, in certain cases, generalization causes some features or parts of features to drop out. This can happen with very small or narrow features. The features display when zoomed in, but as the user zooms out, part or all of an affected feature drops out. The tradeoff for not generalizing features is that the ArcIMS Spatial Server takes longer to process a request.

Attribute Descriptions for IMAGEGENERALIZATION:

Attribute	Usage
mode	Determines whether generalization is turned on. The default is "true", which means generalization is done.

Examples for IMAGEGENERALIZATION:

Example 1: When in SERVICEINFO.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-10216344.0958949" miny="13131205.1055646"
maxx="-10214678.6895209" maxy="13137124.071447" name="Initial_Extent"
/>
        <MAPUNITS units="meters" />
        <IMAGEGENERALIZATION mode="true" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="streets"
id="0">
        <FCLASS type="line"></FCLASS>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Example 2: When used in CONFIG.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-10216344.095894964" miny="13131205.105564624"
maxx="-10214678.689520996" maxy="13137124.071447073"
name="Initial_Extent" />
        <MAPUNITS units="meters" />
        <IMAGEGENERALIZATION mode="true" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to data>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="roads" visible="true" id="0">
        <DATASET name="roads" type="line" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLELINESYMBOL width="1" captype="round"
color="27,127,127" />
```

```
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

IMAGELIMIT

Used in: **RESPONSE**
Servers: Image ArcMap
Parent elements: **ENVIRONMENT**

<IMAGELIMIT

pixelcount ***="integer"***
>

No Child Elements
</IMAGELIMIT >

Bold: Attribute or child element is required.

Description:

Defines maximum allowable size of output map in total number of pixels.

Restrictions:

None

Notes:

- IMAGELIMIT is set when an Image or ArcMap Image Service is started. By default, an image is limited in size to 4 MB, which corresponds to 1,048,576 pixels. Changes can be made to the image size using ArcIMS Administrator. For more information, see *Using ArcIMS*.

Attribute Descriptions for IMAGELIMIT:

Attribute	Usage
pixelcount	Represents the maximum number of pixels allowed in a map image. The calculation is made by multiplying the width times the height. For example, an image 400 x 600 in size contains 240000 pixels.

Examples for IMAGELIMIT:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
```

```
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-71.0718204242754" miny="42.368904975182"
maxx="-71.0475995680561" maxy="42.3869647980717" name="Initial_Extent"
/>
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="Streets"
id="1" maxscale="0.00004703130261735 83">
        <FCLASS type="line"></FCLASS>
        <EXTENSION type="Geocode" >
          <GCSTYLE name="USAddressZ" />
        </EXTENSION>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

IMAGEPROPERTIES

Used in: **CONFIG** **REQUEST**
Servers: Image
Parent elements: **LAYER**

```
<IMAGEPROPERTIES

    transcolor ="0,0,0 - 255,255,255"
    transparency ="0.0 - 1.0"  [1.0]
>

    No Child Elements
</IMAGEPROPERTIES >
```

Description:

Sets the opaqueness of a raster layer such as a TIFF image or sets a color as transparent.

Restrictions:

- In map configuration files, either *transcolor* or *transparency* must be present. It is also acceptable to have both attributes present. In viewer configuration files, only transparency is used.
- Valid with Image Server only.

Notes:

- In map configuration files, *transparency* is applied to the entire image and *transcolor* is 100 percent transparent. The attribute *transcolor* can be a different color than the transparent color used for BACKGROUND.
- When transparency is set on an Image Service layer in a viewer configuration file, the transparency is set for the entire layer, not for the individual sublayers.

Attribute Descriptions for IMAGEPROPERTIES:

Attribute	Usage
transcolor	Transparency color using RGB values. White is 255,255,255.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.

Examples for IMAGEPROPERTIES:

Example 1: When in a map configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="477075.998116" miny="3761051.655341"
maxx="495210.599161" maxy="3773575.12005" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
        <BACKGROUND transcolor="255,255,255" color="255,255,255" />
      </PROPERTIES>
      <WORKSPACES>
        <IMAGEWORKSPACE directory="F:\Data\Redlands1" name="jai_ws-0"
/>
      </WORKSPACES>
      <LAYER type="image" name="redlands.sid" visible="true" id="0">
      <DATASET name="redlands.sid" type="image" workspace="jai_ws-0" />
      <IMAGEPROPERTIES transparency="0.5" transcolor="255,255,255"
/>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When in a viewer configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-74.07187110438655" miny="40.6899519190673"
maxx="-73.81759897554464" maxy="40.883113267364" name="Initial_Extent"
/>
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
      <IMAGESERVERWORKSPACE name="mapper_ws-0"
url="http://mycomputer.domain.com/servlet/com.esri.esrimap.Esrimap"
service="nyc" />
      </WORKSPACES>
      <LAYER type="image" name="nyc" visible="true" id="0">
      <DATASET name="nyc" type="image" workspace="mapper_ws-0" />
```

```
      <IMAGEPROPERTIES transparency="0.5" />
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

IMAGESERVERWORKSPACE

Used in: CONFIG
Parent elements: WORKSPACES

```
<IMAGESERVERWORKSPACE

    name ="string"
    service ="string"
    url ="string"
>

    No Child Elements
</IMAGESERVERWORKSPACE >
```

Bold: Attribute or child element is required.

Description:

Specifies a workspace for an ArcIMS Image Service.

Restrictions:

- Can only be used in viewer configuration files. It cannot be used in a map configuration file.

Notes:

None

Attribute Descriptions for IMAGESERVERWORKSPACE:

Attribute	Usage
name	Workspace name. Must be unique among all data sources.
service	ArcIMS service name.
url	URL points to location of ArcIMS servlet connector (servlet/com.esri.esrimap.Esrimap).

Examples for IMAGESERVERWORKSPACE:

Example 1: When in a viewer configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-89.9747543334961"
maxx="179.9423828125" maxy="90.0" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <IMAGESERVERWORKSPACE name="mapper_ws-0"
url="http://mycomputer.domain.com/servlet/com.esri.esrimap.Esrimap"
service="background" />
      </WORKSPACES>
      <LAYER type="image" name="background" visible="true" id="0">
        <DATASET name="background" type="image" workspace="mapper_ws-0"
/>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

IMAGESIZE

Used in: REQUEST
Servers: Image Extract ArcMap
Parent elements: PROPERTIES

<IMAGESIZE

*When parent element is **GET_EXTRACT, GET_LAYOUT:***
height **="1 - NNN"**
width **="1 - NNN"**

*When parent element is **GET_IMAGE when using ArcMap Server:***
height **="1 - NNN"**
width **="1 - NNN"**
dpi ="1 - NNN"

*When parent element is **GET_IMAGE when using Image Server:***
height **="1 - NNN"**
width **="1 - NNN"**
dpi ="1 - NNN"
printheight ="1 - NNN" **[Same as height]**
printwidth ="1 - NNN" **[Same as width]**
scalesymbols ="true | false" **[false]**
>

No Child Elements
</IMAGESIZE >

Bold: Attribute or child element is required.

Description:

When used with GET_IMAGE or GET_LAYOUT, defines width and height of output map. When used with GET_EXTRACT, defines which layers should be extracted if scale dependencies are present.

Restrictions:

- Used only with Image, Extract, and ArcMap Servers.

Notes:

- In a GET_EXTRACT request, IMAGESIZE is used to calculate which layers should be extracted based on any scale dependencies. If a layer is out of range based on the scale, it will not be extracted.
- The image output map size can be determined three different ways:
  - By using *height* and *width* alone.

- By using *printwidth* and *printheight* along with *height* and *width*. (Image Services only)
- By using *width*, *height*, and *dpi*. (Image and ArcMap Image Services)

The first method is the default, but this method does not take scale dependencies into consideration. If the size of the image changes, but the envelope remains the same, the scale of the map changes. The result is that data content and layer rendering may change as the width and height increase or decrease. To avoid changing the scale as a map is increased or decreased in size, the second two options do take scale dependencies into consideration. Both options produce the same results, but the method for calculating the new map differs. In the second option, calculations are made using pixels. In the third option, calculations are made using dots per inch (dpi). A more thorough explanation of using these attributes is in the IMAGESIZE section of Using GET_IMAGE and IMAGE with Image Services or Using GET_IMAGE and IMAGE with ArcMap Image Services .

- When printheight and printwidth are used with Image Services, all symbols and labels are resized to stay in the same relative proportion as the map increases or decreases in size. For example, in a map that has been doubled in size, the symbols and labels also double in size.

Attribute Descriptions for IMAGESIZE:

*When parent element is **GET_EXTRACT, GET_LAYOUT:***

Attribute	Usage	
height	Image height in pixels.	
width	Image width in pixels.	

*When parent element is **GET_IMAGE when using ArcMap Server:***

Attribute	Usage	
dpi	Dots per inch (dpi).	
height	Image height in pixels.	
width	Image width in pixels.	

*When parent element is **GET_IMAGE when using Image Server:***

Attribute	Usage
dpi	Dots per inch (dpi).
height	Image height in pixels.
printheight	Height of the output image in pixels.
printwidth	Width of the output image in pixels.
scalesymbols	Determines whether to increase or decrease symbology in proportion to a map image as the image increases or decreases in size.
width	Image width in pixels.

Examples for IMAGESIZE:

Example 1: When in a GET_IMAGE request using printwidth and printheight.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_IMAGE>
    <PROPERTIES>
      <IMAGESIZE height="640" width="480" printheight="800"
printwidth="600" />
    </PROPERTIES>
  </GET_IMAGE>
</REQUEST>
</ARCXML>
```

Example 2: When in a GET_IMAGE request using dpi.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-73.985" miny="40.756" maxx="-73.972"
maxy="40.765" />
        <IMAGESIZE width="350" height="245" dpi="134"
scalesymbols="true" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

IMAGEWORKSPACE

Used in: **CONFIG** **REQUEST**
Servers: Image Query Feature
Parent elements: **WORKSPACES**

<IMAGEWORKSPACE

directory ="***string***"
name ="***string***"
>

No Child Elements
</IMAGEWORKSPACE >

Bold: Attribute or child element is required.

Description:

Specifies a workspace for image files.

Restrictions:

- Must refer to an existing data source.
- Not valid with ArcMap Server.

Notes:

- Valid in both viewer configuration files and map configuration files.
- The directory specified for IMAGEWORKSPACE varies depending on how images are accessed.

Image Access Method	IMAGEWORKSPACE Location	DATASET Layer Name
Specify by name.	Attribute *directory* points to location of specified image.	Name of image including its extension.
Use all images in a directory. Images in the same directory automatically tile if they use the same coordinate projection and are drawn when they are within the extent requested.	Attribute *directory* points to location of the group of images.	Use *ImageDirectory for the name: *name="*ImageDirectory"*
Use an ArcView GIS image catalog.	Attribute *directory* points to location of catalog, not images.	Name of the image catalog DBF file. For instance, if catalog is

		named imagecat.dbf, use *name="imagecat.dbf"*.
Add a GRID.	A GRID has two directories: one for the GRID data and one for the INFO files. Both these directories should be grouped together under a parent directory. The *directory* attribute points to the parent directory above the INFO and GRID directories. A *.clr file can be included to color the GRID. This file should have the same name as the GRID and be included in the parent directory above the INFO and GRID directories.	Name of directory that contains GRID data; for a GRID named WorldImage, use *name="WorldImage"*.

- For a listing of supported images, see *Using ArcIMS*.

Attribute Descriptions for IMAGEWORKSPACE:

Attribute	Usage
directory	Directory containing images or image catalog. UNC pathnames can be used (\\myComputer\imagedirectory). See Notes section for more information.
name	Workspace name. Must be unique among all data sources.

Examples for IMAGEWORKSPACE:

Example 1: When specifying one image by name.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="192837" miny="3769109" maxx="197809"
maxy="3773771" name="Initial Extent" />
```

```
      <MAPUNITS units="meters" />
      </PROPERTIES>
      <WORKSPACES>
      <IMAGEWORKSPACE directory="<path to data>" name="jai_ws-0" />
      </WORKSPACES>
      <LAYER type="image" name="reno.sid" visible="true" id="0">
      <DATASET name="reno.sid" type="image" workspace="jai_ws-0" />
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When specifying multiple images in a directory.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="0" miny="0.0" maxx="891.0" maxy="1000.0"
name="Initial_Extent" />
      <MAPUNITS units="meters" />
      </PROPERTIES>
      <WORKSPACES>
      <IMAGEWORKSPACE directory="<path to data>" name="jai_ws-0" />
      </WORKSPACES>
      <LAYER type="image" name="Sierra" visible="false" id="0">
      <DATASET name="*Image" type="image" workspace="jai_ws-0" />
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 3: When specifying an image catalog.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
       <ENVELOPE minx="0" miny="0" maxx="200000" maxy="200000"
name="Initial_Extent"/>
       <MAPUNITS units="meters" />
      </PROPERTIES>
      <WORKSPACES>
```

```
       <IMAGEWORKSPACE name="jai_ws-15" directory="<path to image
catalog dbf file>"/>
      </WORKSPACES>
      <LAYER type="image" name="mammoth.dbf" visible="true" id="0">
       <DATASET name="mammoth.dbf" type="image" workspace="jai_ws-15"
/>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 4: When specifying a GRID.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
       <ENVELOPE minx="-2006008" miny="-1240677" maxx="-1993628"
maxy="-1256187" name="Initial_Extent" />
       <MAPUNITS units="meters" />
      </PROPERTIES>
      <WORKSPACES>
       <IMAGEWORKSPACE directory="<path to directory above INFO
directory>" name="jai_ws-15" />
      </WORKSPACES>
      <LAYER type="image" name="Mt St. Helens" visible="true" id="0">
       <DATASET name="helens" type="image" workspace="jai_ws-15" />
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

LAYER

Used in: CONFIG REQUEST RESPONSE
Servers: Image Query Feature Extract Geocode ArcMap
Parent elements: GET_FEATURES GET_GEOCODE GET_IMAGE LAYERS MAP

<LAYER

*When parent element is **GET_FEATURES, GET_GEOCODE:***
id ="***string***"

*When parent element is **LAYERS in a GET_IMAGE request or map configuration file:***
id ="***string***"

*When parent element is **LAYERS in an IMAGE response:***
featurecount ="***string***"
id ="***string***"
name ="***string***"

*When parent element is **MAP, GET_IMAGE, GET_EXTRACT:***
id ="***string***"
type ="**featureclass | acetate | image**"
maxscale ="*string*"
minscale ="*string*"
name ="*string*"
visible ="true | false" **[true]**
>

*When parent element is **GET_FEATURES, GET_GEOCODE:***
No Child Elements

*When parent element is **GET_IMAGE and LAYER type="featureclass" or "image" - ArcMap Server:***
<DATASET... />
<QUERY... /> *[Either SPATIALQUERY or QUERY but not both]*
<SPATIALQUERY... /> *[Either SPATIALQUERY or QUERY but not both]*

*When parent element is **LAYERS:***
No Child Elements

*When parent element is **MAP or GET_IMAGE and LAYER type="acetate" - Image and ArcMap Servers:***
(m) <OBJECT... />

*When parent element is **MAP or GET_IMAGE and LAYER type="image" - Image Server:***

<**DATASET... /**>
<COORDSYS... />
<IMAGEPROPERTIES... />
<RASTER_RENDERER... /> *[Only valid with MAP]*

*When parent element is **MAP, GET_IMAGE, or GET_EXTRACT and LAYER type="featureclass" - Image and Extract Servers:***

<**DATASET... /**>
<COORDSYS... />
<DENSIFY... />
<QUERY... /> *[Either SPATIALQUERY or QUERY but not both]*
<SPATIALQUERY... /> *[Either SPATIALQUERY or QUERY but not both]*
(m) <EXTENSION... /> *[Only valid with MAP]*
<GROUPRENDERER... /> *[Or]*
<SCALEDEPENDENTRENDERER... /> *[Or]*
<SIMPLELABELRENDERER... /> *[Or]*
<SIMPLERENDERER... /> *[Or]*
<VALUEMAPLABELRENDERER... /> *[Or]*
<VALUEMAPRENDERER... /> *[Or]*

</LAYER >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

- When parent element is MAP, LAYER defines a map layer. Layers of type featureclass use point, line, and polygon data from shapefiles or ArcSDE. Layers of type image are used for displaying images. Layers of type acetate are used to display additional objects on the map.
- In a GET_IMAGE or GET_EXTRACT request, LAYER can be used two ways. It can be used to add a dynamic layer to an existing Image Service. It can also be used to add an existing Image Service layer in a modified format. This latter functionality is most commonly used to show subsets of an existing layer such as selected features.
- Used in a GET_GEOCODE request to identify which layer is to be geocoded.
- Used in a GET_FEATURES request to identify from which layer to get feature information.
- Used with LAYERS in LEGEND to identify which layers in a service or request should not be included in the legend.
- Used in an IMAGE response to give information on the returned vector, image, and acetate layers including name, ID, and number of features in the map.

Restrictions:

- ID for each layer must be unique within the service and within any layers added dynamically in a request.
- In a request to an ArcMap Image Service, LAYER cannot be used to add new data in a dynamic layer.
- Use only one renderer at a time.
- Identify and query cannot be made on dynamic layers.
- EXTENSION and RASTER_RENDERER are valid only when parent element is MAP, and not in a request.

Notes:

- In order to add a dynamic layer with new data in a GET_IMAGE request, the attribute *dynamic* must be set to true in the MAP element in a map configuration file. Valid only with Image Server.
- In a GET_IMAGE or GET_EXTRACT request, LAYER is used to create a new layer. Use LAYERDEF to change an existing layer's rendering or to turn the layer on or off. For more information about using LAYER in a request, including more examples, see Using GET_IMAGE and IMAGE with Image Services, Using GET_IMAGE and IMAGE with ArcMap Image Services, and Using GET_EXTRACT and EXTRACT.
- For ArcSDE layers, DATASET must be defined before renderer.
- Many OBJECTs can be defined in one acetate layer.
- Scales can be set in ArcXML using a relative scale or by calculating the number of map units per pixel. A relative scale represents the scale in a ratio such as 1:24000. In this example, 1 meter equals 24000 meters, or 1 inch equals 24000 inches. When using relative scale, always use a colon (:) between the two values.

  Map units per pixel refers to the number of meters, feet, or decimal degrees represented by one pixel in a map. To convert from a relative scale to map units per pixel, the size of a pixel must first be calculated. The formula for finding the number of meters in a pixel is 0.0254 / dpi where:
  - The value 0.0254 is the number of meters in an inch.
  - Dpi is the dpi set in the ArcIMS service or request. If no dpi is set in the service or request, the dpi is assumed to be 96.

  As an example of calculating pixel size, if the dpi is 96, the pixel size is 0.0254 / 96 or 0.000265 m.

  Once the pixel size is known, the relative scale can be converted to map units per pixel:

  1. If the scale is in **meters**. To calculate the number of meters per pixel, take the relative scale and multiply by the pixel size (0.000265). For example, if the relative scale is 1:24000, then the number of meters per pixel is 24000 * 0.000265, or 6.36 meters.
  2. If the scale is in **feet**. Do the calculation for meters (#1). Multiply the result by 3.28 (the number of feet in a meter). For example, if the number of meters per pixel is 6.36, the number of feet per pixel is 6.36 * 3.28, or 20.86 feet.

3. If the scale is in **decimal degrees**. For these calculations, the earth is assumed to be an exact circle with a circumference of 40030.174 km. One degree is 111.195 km (40030.174/360 degrees), or 111195 meters. To calculate the number of degrees, first do the calculation for meters (#1). Next, divide the result by 111195. For example, if the number of meters per pixel is 6.36, the number of degrees per pixel is 6.36 / 111195, or 0.0000571968.

Attribute Descriptions for LAYER:

*When parent element is **GET_FEATURES, GET_GEOCODE:***

Attribute	Usage
id	ID as specified in map configuration file. For a request to be valid, the ID used in the request must have a match in the map configuration file.

*When parent element is **LAYERS in a GET_IMAGE request or map configuration file:***

Attribute	Usage
id	ID as specified in map configuration file or the request. For a legend list to be valid, the ID used must have a match in the map configuration file or the request.

*When parent element is **LAYERS in an IMAGE response:***

Attribute	Usage
featurecount	For shapefile and ArcSDE layers, counts the number of features shown on the map. For image layers, counts the number of images present in the map. For acetate layers, counts the number of times OBJECT is used in a layer.
id	ID as specified in map configuration file or request.
name	Name as specified in map configuration file or request.

*When parent element is **MAP, GET_IMAGE, GET_EXTRACT:***

Attribute	Usage
id	Unique ID for a layer. The ID can be any combination of alpha and numeric characters. If used in a request to extract data, the characters must be in English and valid for a filename unless the extract EXTENSION for the layer is used.
maxscale	Maximum scale to display map using a relative scale such as 1:24000. Scale can also be calculated as the number of map units per pixel.
minscale	Minimum scale to display map using a relative scale such as 1:24000. Scale can also be calculated as the number of map units per pixel.
name	Layer name. Can be an alias.
type	Specifies layer type. Use "featureclass" for shapefiles and ArcSDE vector layers. Use "image" for raster image files, GRIDs, and ArcSDE raster layers. Use "acetate" for adding graphics on top of the map.
visible	Specifies layer visibility.

Examples for LAYER:

Example 1: When in CONFIG.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.21" miny="18.92" maxx="-66.96"
maxy="71.41" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CITIES" visible="true" id="2">
      <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLEMARKERSYMBOL type="square" width="5" />
      </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: Map configuration file where layers with ID of "0" and "1" are removed from the legend list.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP dynamic="true" >
      <PROPERTIES>
      <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      <LEGEND title="Legend" font="Arial" autoextend="true" columns="1"
width="170" height="300" backgroundcolor="255,255,0" >
        <LAYERS>
          <LAYER id="0"/>
          <LAYER id="1"/>
        </LAYERS>
      </LEGEND>
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>" />
        <SHAPEWORKSPACE name="shp_ws-2" directory="<path to USA
ESRIDATA>" />
        <SHAPEWORKSPACE name="shp_ws-3" directory="<path to CANADA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="WORLD30" visible="true" id="0">
        <DATASET name="WORLD30" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="0,153,255"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="1">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,255,153"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="United States" visible="true"
id="2">
        <DATASET name="STATES" type="polygon" workspace="shp_ws-2" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Canada" visible="true" id="3">
        <DATASET name="province" type="polygon" workspace="shp_ws-3" />
        <SIMPLERENDERER>
```

```
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="0,153,0" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="World Cities" visible="true"
id="4">
        <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="102,0,102" width="8.0" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 3: When used with LAYERS to remove layers with ID of "0" and "1" from the legend list.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <LEGEND title="Legend" font="Arial" autoextend="true"
columns="1" width="170" height="300" backgroundcolor="255,255,0" >
          <LAYERS>
            <LAYER id="0"/>
            <LAYER id="1"/>
          </LAYERS>
        </LEGEND>
        <DRAW map="false"/>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 4: Identifying a layer by its ID in a GET_FEATURES request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_FEATURES featurelimit="25" beginrecord="0" outputmode="xml"
geometry="false" envelope="true" compact="true">
    <LAYER id="4" />
    <SPATIALQUERY subfields="#ALL#" where="NAME  =  'Los Angeles'" >
    </SPATIALQUERY>
  </GET_FEATURES>
</REQUEST>
</ARCXML>
```

Example 5: IMAGE response showing details on layers in the requested ArcIMS service.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-180" miny="-135" maxx="180" maxy="135" />
      <LAYERS>
        <LAYER name="CNTRY94" id="0" featurecount="165" />
        <LAYER name="STATES" id="1" featurecount="51" />
        <LAYER name="province" id="2" featurecount="12" />
      </LAYERS>
      <OUTPUT file="c:\arcims\output\world_MYCOMPUTER3633699.jpg"
url="http://mycomputer.domain.com/output/world_MYCOMPUTER3633699.jpg"
/>
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

LAYERDEF

Used in: **CONFIG** **REQUEST**
Servers: Image Extract ArcMap
Parent elements: **LAYERLIST** **OVERVIEWMAP**

<LAYERDEF

*When parent element is **LAYERLIST in GET_IMAGE or GET_EXTRACT request:***
id ="*string*"
name ="*string*"
visible ="true | false" **[As defined in map configuration file]**

*When parent element is **OVERVIEWMAP in a viewer configuration file:***
name ="*string*"
>

*When parent element is **LAYERLIST in GET_EXTRACT request:***
<QUERY... /> *[Either SPATIALQUERY or QUERY but not both]*
<SPATIALQUERY... /> *[Either SPATIALQUERY or QUERY but not both]*

*When parent element is **LAYERLIST in GET_IMAGE request - ArcMap Server:***
<QUERY... /> *[Either SPATIALQUERY or QUERY but not both]*
<SPATIALQUERY... /> *[Either SPATIALQUERY or QUERY but not both]*

*When parent element is **LAYERLIST in GET_IMAGE request - Image Server:***
<QUERY... /> *[Either SPATIALQUERY or QUERY but not both]*
<SPATIALQUERY... /> *[Either SPATIALQUERY or QUERY but not both]*
<GROUPRENDERER... /> *[Or]*
<SCALEDEPENDENTRENDERER... /> *[Or]*
<SIMPLELABELRENDERER... /> *[Or]*
<SIMPLERENDERER... /> *[Or]*
<VALUEMAPLABELRENDERER... /> *[Or]*
<VALUEMAPRENDERER... /> *[Or]*

*When parent element is **OVERVIEWMAP in a viewer configuration file:***
No Child Elements

</LAYERDEF >

Bold: Attribute or child element is required.

Description:

- When parent element is LAYERLIST in a request, LAYERDEF is used to define new properties for an existing layer in an Image Service or in the request.

Changes include modifying the rendering and setting a filter using a query. It is also used to turn a layer on or off.
- When the parent element is OVERVIEWMAP in a viewer configuration file, LAYERDEF is used to set which layers are visible in the overview map.

Restrictions:

- Used only in GET_IMAGE and GET_EXTRACT requests.
- If using renderer child elements, only one renderer can be used at a time.
- When used with GET_EXTRACT, the resulting shapefile names by default are based on the LAYERDEF *id*. Characters are limited to those that are also valid for a filename. An alternative to using the LAYERDEF *id* is to use EXTRACTPARAMS for a LAYER in a map configuration file.

Notes:

- New queries added are restricted by any queries that are already defined for this layer in the ArcIMS service.
- For more information about using LAYERDEF in a request, including more examples, see Using GET_IMAGE and IMAGE with Image Services, Using GET_IMAGE and IMAGE with ArcMap Image Services, and Using GET_EXTRACT and EXTRACT.

Attribute Descriptions for LAYERDEF:

*When parent element is **LAYERLIST in GET_IMAGE or GET_EXTRACT request:***

Attribute	Usage
id	Reference to unique layer ID as defined in map configuration file or in a dynamic layer in the request.
name	Reference to layer name as defined in map configuration file or in a dynamic layer in the request.
visible	Turns layer on or off.

*When parent element is **OVERVIEWMAP in a viewer configuration file:***

Attribute	Usage
name	Reference to layer name.

Examples for LAYERDEF:

Example 1: When in a GET_IMAGE request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
  <GET_IMAGE>
    <PROPERTIES>
      <LAYERLIST>
        <LAYERDEF id="1">
          <SIMPLERENDERER>
            <SIMPLEMARKERSYMBOL size="16" color="0,0,0" />
          </SIMPLERENDERER>
         <SPATIALQUERY>
            <SPATIALFILTER relation="area_intersection">
            <ENVELOPE maxy="60" maxx="60" miny="0" minx="0" />
         </SPATIALFILTER>
         </SPATIALQUERY>
      </LAYERDEF>
      </LAYERLIST>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
   </PROPERTIES>
  </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 2: When used in OVERVIEWMAP.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-116.016078" miny="36.252371" maxx="-100.855887"
maxy="46.622450" name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
      <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="STATES" visible="true" id="0">
      <DATASET name="STATES" type="polygon" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLEPOLYGONSYMBOL fillcolor="255,0,0" />
      </SIMPLERENDERER>
      </LAYER>
    </MAP>
    <OVERVIEWMAP backgroundcolor="255,255,255" framefillcolor="255,0,0"
frameoutlinecolor="255,0,0" zoomfactor="4.0">
```

```
      <LAYERDEF name="STATES" />
    </OVERVIEWMAP>
  </CONFIG>
</ARCXML>
```

LAYERINFO

Used in: RESPONSE
Servers: Image Query Feature Extract Geocode ArcMap
Parent elements: DATAFRAMEINFO SERVICEINFO

<LAYERINFO

id ="***string***"
type ="**featureclass | image | acetate**"
maxscale ="*string*"
minscale ="*string*"
name ="*string*"
visible ="true | false"
>

*When parent element is **LAYERINFO and type is acetate:***
No Child Elements

*When parent element is **LAYERINFO and type is featureclass for ArcMap Image Services:***
<FCLASS... />

*When parent element is **LAYERINFO and type is featureclass for Image or Feature Services:***
<FCLASS... />
<EXTENSION... />
<GROUPRENDERER... />
<SCALEDEPENDENTRENDERER... />
<SIMPLELABELRENDERER... />
<SIMPLERENDERER... />
<VALUEMAPLABELRENDERER... />
<VALUEMAPRENDERER... />

*When parent element is **LAYERINFO and type is image:***
<ENVELOPE... />

</LAYERINFO >

Bold: Attribute or child element is required.

Description:

Provides the layout size for the layout of an ArcMap Image Service.

Restrictions:

None

Notes:

- Scales in LAYERINFO are based on the number of map units per pixel. In requests and the map configuration file, scales can be set using a relative scale ratio such as 1:24000. In this example, 1 meter equals 24000 meters or 1 inch equals 24000 inches.

  Map units per pixel refers to the number of meters, feet, or decimal degrees represented by one pixel in a map. To convert from a relative scale to map units per pixel, the size of a pixel must first be calculated. The formula for finding the number of meters in a pixel is 0.0254 / dpi where:

1. The value 0.0254 is the number of meters in an inch.
2. Dpi is the dpi set in the service or request. If no dpi is set in the ArcIMS service or request, the dpi is assumed to be 96.

   As an example of calculating pixel size, if the dpi is 96, the pixel size is 0.0254 / 96 or 0.000265 m.

   Once the pixel size is known, the relative scale can be converted to map units per pixel:

   3. If the scale is in **meters**. To calculate the number of meters per pixel, take the relative scale and multiply by the pixel size (0.000265). For example, if the relative scale is 1:24000, then the number of meters per pixel is 24000 * 0.000265, or 6.36 meters.
   4. If the scale is in **feet**. Do the calculation for meters (#1). Multiply the result by 3.28 (the number of feet in a meter). For example, if the number of meters per pixel is 6.36, the number of feet per pixel is 6.36 * 3.28, or 20.86 feet.
   5. If the scale is in **decimal degrees**. For these calculations, the earth is assumed to be an exact circle with a circumference of 40030.174 km. One degree is 111.195 km (40030.174/360 degrees) or 111195 meters. To calculate the number of degrees, first do the calculation for meters (#1). Next, divide the result by 111195. For example, if the number of meters per pixel is 6.36, the number of degrees per pixel is 6.36 / 111195, or 0.0000571968.

Attribute Descriptions for LAYERINFO:

Attribute	Usage
id	Reference to unique layer ID as defined in map configuration file or ArcMap document.
maxscale	Maximum scale to display map in map units per pixel. Does not apply unless maxscale is set in the service.
minscale	Minimum scale to display map in map units per pixel. Does not apply unless minscale is set in the service.
name	Layer name as defined in map configuration file or ArcMap document.
type	Shows type of layer as defined in map configuration file or ArcMap document.
visible	Specifies whether or not layer is visible.

Examples for LAYERINFO:

Example 1: When in a SERVICEINFO response.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="Countries"
id="1">
        <FCLASS type="polygon">
          <FIELD name="AREA" type="8" size="12" precision="3" />
          <FIELD name="NAME" type="12" size="40" precision="0" />
          <FIELD name="ABBREVNAME" type="12" size="12" precision="0" />
          <FIELD name="FIPS_CODE" type="12" size="2" precision="0" />
          <FIELD name="WB_CNTRY" type="12" size="3" precision="0" />
          <FIELD name="HYPERLINK" type="12" size="60" precision="0" />
          <FIELD name="#SHAPE#" type="-98" size="0" precision="0" />
          <FIELD name="#ID#" type="-99" size="16" precision="0" />
        </FCLASS>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

LAYERLIST

Used in: REQUEST
Servers: Image Extract ArcMap
Parent elements: PROPERTIES

<LAYERLIST

*When parent element is **GET_EXTRACT:***
nodefault ="true | false" **[false]**

*When parent element is **GET_IMAGE using the ArcMap Server:***
order ="true | false" **[false]**

*When parent element is **GET_IMAGE using the Image Server:***
dynamicfirst ="true | false" **[false]**
nodefault ="true | false" **[false]**
order ="true | false" **[false]**
>

(m) <LAYERDEF... />

</LAYERLIST >

(m): Child element can be used multiple times.

Description:

Defines list of layers to be drawn on the map in a GET_IMAGE request or extracted in a GET_EXTRACT request.

Restrictions:

- GET_IMAGE attributes are valid with Image Server only. They are not valid with ArcMap Server.

Notes:

- By default in a GET_IMAGE request, the layers are drawn in the following order:
    1. Layers in the ArcIMS service in the order they are defined in the service including acetate layers.
    2. Dynamic layers added in the request in the order they are defined in the request.

  When using the Image Server, the layer order can be changed by using the *order* attribute. When *order* is set to true, layers are drawn in the order they are listed in the LAYERLIST. This means that dynamic layers can be intermingled among the layers from the service. Both service layers and dynamic layers must be identified

using LAYERDEF. The LAYER *id* must be unique among all service layers and dynamic layers.

- The attribute *dynamicfirst* is ignored if *nodefault* and/or *order* is set to true.
- For more information about using LAYERLIST in a request, including more examples, see Using GET_IMAGE and IMAGE with Image Services, Using GET_IMAGE and IMAGE with ArcMap Image Services, and Using GET_EXTRACT and EXTRACT.

Attribute Descriptions for LAYERLIST:

*When parent element is **GET_EXTRACT:***

Attribute	Usage
nodefault	When set to "true", only the layers listed in the LAYERLIST are extracted. Only ArcIMS service and dynamic layers of type featureclass are extracted. Acetate and image layers included in the LAYERLIST are not extracted. If *visible* is set to "false" in LAYERDEF, then the layer is not included.

*When parent element is **GET_IMAGE using the ArcMap Server:***

Attribute	Usage
order	When set to "true", only layers in the LAYERLIST are drawn. All layers, including acetate layers, must have a unique ID.

*When parent element is **GET_IMAGE using the Image Server:***

Attribute	Usage
dynamicfirst	When set to "true", draws all dynamic layers before any layers defined in the service.
nodefault	When set to "true", only ArcIMS service, dynamic layers, and acetate layers included in the LAYERLIST are displayed on the map. Service layers are displayed first followed by dynamic layers in the order they appear in the map configuration file and request, respectively. When *nodefault* is "false", all dynamic and service layers are included. In both cases, if *visible* is set to "false" in LAYERDEF, then the layer does not display.
order	When set to "true", draws layers in the order listed in the LAYERDEF elements. Only layers in the LAYERLIST are drawn. Can include ArcIMS service layers, dynamic layers, and acetate layers in any order. All layers, including acetate layers, must have a unique ID.

Examples for LAYERLIST:

Example 1: When specifying the order that layers should be drawn in a GET_IMAGE request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE show="">
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <IMAGESIZE width="800" height="600"/>
        <LAYERLIST dynamicfirst="false" nodefault="false"
order="true">
          <LAYERDEF id="0" visible="true" />
          <LAYERDEF id="333" visible="true" />
          <LAYERDEF id="2" visible="true" />
          <LAYERDEF id="3" visible="true" />
          <LAYERDEF id="1" visible="true" >
            <SIMPLERENDERER>
              <SIMPLEPOLYGONSYMBOL filltransparency="0.5"
filltype="solid" fillcolor="255,255,153" />
              </SIMPLERENDERER>
          </LAYERDEF>
          <LAYERDEF id="4" visible="true" >
            <SPATIALQUERY where="POPULATION &gt; 2000000" >
              <SPATIALFILTER relation="area_intersection">
                <ENVELOPE minx="-14.0" miny="35.0" maxx="33.0"
maxy="64.0" />
              </SPATIALFILTER>
            </SPATIALQUERY>
            <SIMPLERENDERER>
              <SIMPLEMARKERSYMBOL type="star" color="0,155,0"
width="12.0" />
            </SIMPLERENDERER>
          </LAYERDEF>
        </LAYERLIST>
      </PROPERTIES>

      <LAYER type="featureclass" name="new_CNTRY94" id="333" >
        <DATASET fromlayer="1" />
        <SPATIALQUERY where="NAME=&apos;Brazil&apos;" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0" />
        </SIMPLERENDERER>
      </LAYER>

      <LAYER type="acetate" name="WorldText" id="444">
        <OBJECT units="database">
          <TEXT size="32" coords="0 0" label="The World">
            <TEXTMARKERSYMBOL fontstyle="bold" fontsize="32"
font="Arial" fontcolor="0,0,0" glowing="255,255,0" />
          </TEXT>
        </OBJECT>
      </LAYER>
```

```
      <LAYER type="acetate" name="WorldBox" id="555" >
        <OBJECT units="pixel">
          <LINE coords="10 40;400 40;400 80;10 80;10 40">
            <SIMPLELINESYMBOL color="0,0,0" />
          </LINE>
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 2: When in a GET_EXTRACT request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_EXTRACT>
      <ENVIRONMENT>
        <SEPARATORS cs="#" ts="*" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-130" miny="30" maxx="-90" maxy="60" />
        <LAYERLIST>
          <LAYERDEF id="Ocean" visible="true" />
          <LAYERDEF id="Countries" visible="true" />
          <LAYERDEF id="States" visible="true" />
          <LAYERDEF id="Provinces" visible="false" />
          <LAYERDEF id="Cities" visible="true" >
            <SPATIALQUERY>
              <SPATIALFILTER relation="area_intersection">
                <POLYGON>
                  <RING>
                    <COORDS> -125#39*-125#63*-90#63*-90#39*-125#39
</COORDS>
                  </RING>
                </POLYGON>
              </SPATIALFILTER>
            </SPATIALQUERY>
          </LAYERDEF>
        </LAYERLIST>
      </PROPERTIES>
    </GET_EXTRACT>
  </REQUEST>
</ARCXML>
```

LAYERS

Used in: CONFIG REQUEST RESPONSE
Servers: Image
Parent elements: IMAGE LEGEND

<LAYERS >
No Attributes

(m) <LAYER... />

</LAYERS >

(m): Child element can be used multiple times.

Description:

- In a request or map configuration file, LAYERS contains a list of layers that should *not* be included in the legend.
- In a response, LAYERS contains a list of layers returned by the Image Server including layer ID and number of features.

Restrictions:

- When used in a map configuration file or GET_IMAGE request, LAYERS is valid only with Image Services in an ArcIMS HTML Viewer or other HTML implementation.
- When used in an IMAGE response, LAYERS is valid only with Image Services.
- Not valid with ArcMap Server.

Notes:

None

Examples for LAYERS:

Example 1: Map configuration file where layers with ID of "0" and "1" are removed from the legend list.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP dynamic="true" >
      <PROPERTIES>
      <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
name="Initial Extent" />
```

```
      <MAPUNITS units="decimal_degrees" />
      <LEGEND title="Legend" font="Arial" autoextend="true" columns="1"
width="170" height="300" backgroundcolor="255,255,0" >
        <LAYERS>
          <LAYER id="0"/>
          <LAYER id="1"/>
        </LAYERS>
      </LEGEND>
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>" />
        <SHAPEWORKSPACE name="shp_ws-2" directory="<path to USA
ESRIDATA>" />
        <SHAPEWORKSPACE name="shp_ws-3" directory="<path to CANADA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="WORLD30" visible="true" id="0">
        <DATASET name="WORLD30" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="0,153,255"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="1">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,255,153"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="United States" visible="true"
id="2">
        <DATASET name="STATES" type="polygon" workspace="shp_ws-2" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Canada" visible="true" id="3">
        <DATASET name="province" type="polygon" workspace="shp_ws-3" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="0,153,0" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="World Cities" visible="true"
id="4">
        <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="102,0,102" width="8.0" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: GET_IMAGE request where layers with ID of "0" and "1" are

removed from the legend list.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <LEGEND title="Legend" font="Arial" autoextend="true"
columns="1" width="170" height="300" backgroundcolor="255,255,0" >
          <LAYERS>
            <LAYER id="0"/>
            <LAYER id="1"/>
          </LAYERS>
        </LEGEND>
        <DRAW map="false"/>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 3: IMAGE response showing details on layers in the requested MapService.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-180" miny="-135" maxx="180" maxy="135" />
      <LAYERS>
        <LAYER name="CNTRY94" id="0" featurecount="165" />
        <LAYER name="STATES" id="1" featurecount="51" />
        <LAYER name="province" id="2" featurecount="12" />
      </LAYERS>
      <OUTPUT file="c:\arcims\output\world_MYCOMPUTER3633699.jpg"
url="http://mycomputer.domain.com/output/world_MYCOMPUTER3633699.jpg"
/>
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

LAYOUT

Used in: RESPONSE
Servers: ArcMap
Parent elements: RESPONSE

<LAYOUT >
No Attributes

<ENVELOPE... />
<OUTPUT... />

</LAYOUT >

Description:

Contains information for a retrieved ArcMap Image Service layout.

Restrictions:

None

Notes:

- See GET_LAYOUT for request.
- When ENVELOPE is used in a LAYOUT response, the coordinates are in the page coordinates of the layout.
- For more details on using LAYOUT and GET_LAYOUT, including additional examples, see Using GET_LAYOUT and LAYOUT with ArcMap Image Services.

Examples for LAYOUT:

Example 1:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <LAYOUT>
      <ENVELOPE minx="-3.08333333333333" miny="0"
maxx="11.5833333333333" maxy="11" />
      <OUTPUT
url="http://mymachine.domain.com/output/world_MYMACHINE1420161231.JPG"
/>
    </LAYOUT>
  </RESPONSE>
</ARCXML>
```

LAYOUTINFO

Used in: **RESPONSE**
Servers: ArcMap
Parent elements: **SERVICEINFO**

```
<LAYOUTINFO

    pageunits ="inches | feet | yards | centimeters | millimeters | decimeters | meters |
points"
>

    <ENVELOPE... />

</LAYOUTINFO >
```

Description:

Provides the initial envelope and units for an ArcMap Image Service that includes a layout.

Restrictions:

None

Notes:

- The ENVELOPE values for LAYOUTINFO are in page units rather than map units.

Attribute Descriptions for LAYOUTINFO:

Attribute	Usage
pageunits	Units for layout page.

Examples for LAYOUTINFO:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US"/>
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular"/>
        <SEPARATORS cs=" " ts=";"/><SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576"/>
        <CAPABILITIES forbidden=""/>
      </ENVIRONMENT>
```

```
      <LAYOUTINFO pageunits="inches">
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
      </LAYOUTINFO>
      <DATAFRAMEINFO name="Layers">
        <PROPERTIES>
          <FEATURECOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
          <FILTERCOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
          <MAPUNITS units="decimal_degrees"/>
          <BACKGROUND color="255,255,255"/>
          <ENVELOPE minx="-127.714285386824" miny="-6.58527935381"
maxx="-63.9877554315853" maxy="81.1449102179015" name="Initial_Extent"
/>
        </PROPERTIES>
        <LAYERINFO type="featureclass" name="states" id="2"
visible="true">
          <FCLASS type="polygon"></FCLASS>
        </LAYERINFO>
        <LAYERINFO type="featureclass" name="rivers" id="1"
visible="true">
          <FCLASS type="line"></FCLASS>
        </LAYERINFO>
        <LAYERINFO type="featureclass" name="cities" id="0"
visible="true">
          <FCLASS type="point"></FCLASS>
        </LAYERINFO>
      </DATAFRAMEINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

LEGEND

Used in: CONFIG REQUEST RESPONSE
Servers: Image ArcMap
Parent elements: IMAGE PROPERTIES

<LEGEND

*When parent element is **IMAGE:***
file ="***string***"
url ="***string***"

*When parent element is **PROPERTIES when using ArcMap Server:***
autoextend ="true | false" **[false]**
backgroundcolor ="0,0,0 - 255,255,255"
display ="true | false" **[true]**
font ="Any system font" **[Arial]**
layerfontsize ="*integer*" **[10]**
swatchheight ="*integer*" **[14]**
swatchwidth ="*integer*" **[18]**
title ="*string*"
titlefontsize ="*integer*" **[12]**
valuefontsize ="*integer*" **[8]**

*When parent element is **PROPERTIES when using Image Server:***
anitaliasing ="true | false" **[true]**
autoextend ="true | false" **[false]**
backgroundcolor ="0,0,0 - 255,255,255"
cansplit ="true | false" **[false]**
cellspacing ="*integer*" **[2]**
columns ="*integer*" **[1]**
display ="true | false" **[true]**
font ="Any system font" **[Arial]**
height ="1 - NNN" **[300]**
layerfontsize ="*integer*" **[10]**
reverseorder ="true | false" **[false]**
splittext ="*string*" **[(cont)]**
swatchheight ="*integer*" **[14]**
swatchwidth ="*integer*" **[18]**
title ="*string*"
titlefontsize ="*integer*" **[12]**
transcolor ="0,0,0 - 255,255,255"
valuefontsize ="*integer*" **[8]**
width ="1 - NNN" **[125]**
>

*When parent element is **IMAGE:***
No Child Elements

*When parent element is **PROPERTIES when using ArcMap Server:***
No Child Elements

*When parent element is **PROPERTIES when using Image Server:***
(m) <LAYERS... />

</LEGEND >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Defines the map's legend when using Image or ArcMap Image Services.

Restrictions:

- When using an ArcIMS HTML Viewer or any other HTML implementation using the ArcIMS Servlet Connector, the LEGEND element is restricted in an IMAGE response. The attribute *file* is not returned.

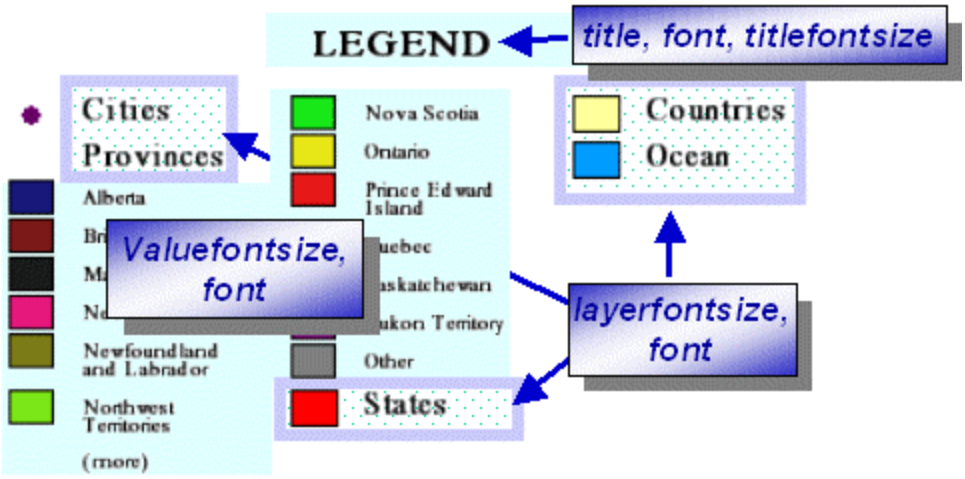
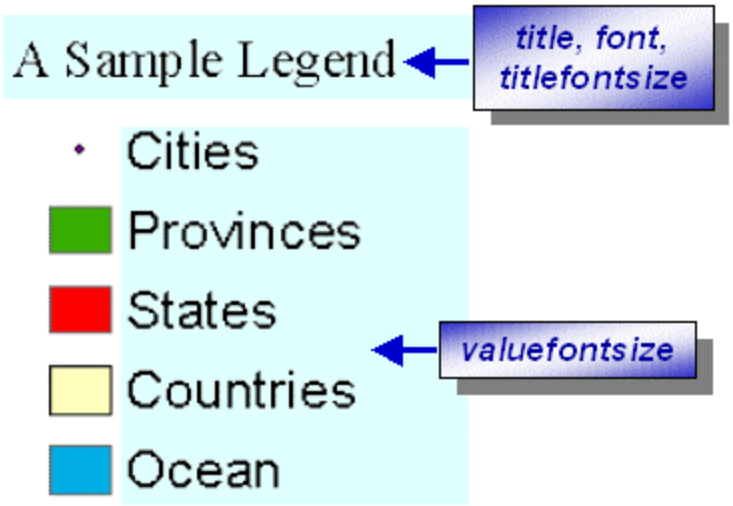
  This restriction can be lifted by setting the property *spatialServer.AllowResponsePath* to true in esrimap_prop. This property file is found in the servlet engine directory. For more information on the location of esrimap_prop and its properties, see the ArcIMS Help.

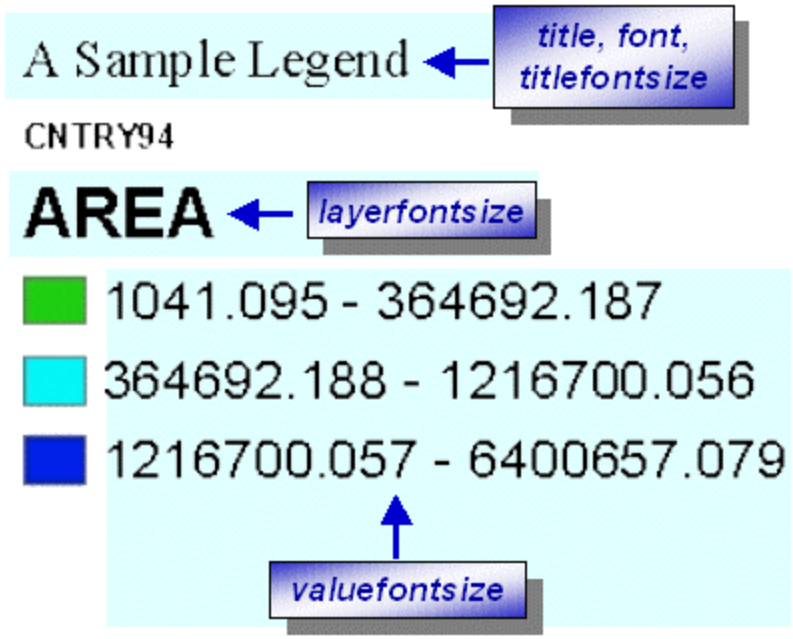
  This restriction applies only when the ArcIMS Servlet Connector is used. It does not apply to the ActiveX, ColdFusion, or Java Connectors.

- Valid only with Image or ArcMap Image Services in an ArcIMS HTML Viewer or other HTML implementation.

Notes:

- If LEGEND is defined in a map configuration file, then a legend is always generated during a GET_IMAGE request even if no legend is defined in the request.
- To remove layers from the legend list, use LAYERS.
- The following table shows which attributes are available for setting the size and font for the title and individual layers. The usage of attributes differs between Image and ArcMap Image Services.

	Attributes available when using an **Image Service** and no layers contain a value map.
	Attributes available when using an **Image Service** and at least one layer contains a value map.
	Attributes available when using an **ArcMap Image Service** and no layers contain a value map.



Attributes available when using an **ArcMap Image Service** and at least one layer contains a value map.

- For a comprehensive overview of using LEGEND, including more examples, see Using GET_IMAGE and IMAGE with Image Services or Using GET_IMAGE and IMAGE with ArcMap Image Services.

Attribute Descriptions for LEGEND:

*When parent element is **IMAGE:***

Attribute	Usage
file	Full pathname and filename for location of legend image generated by the ArcIMS Spatial Server. Image is written to the ArcIMS Spatial Server's output directory by default. UNC pathnames are valid (\\myComputer\arcims\output).
url	URL used by client to retrieve legend image.

*When parent element is **PROPERTIES when using ArcMap Server:***

Attribute	Usage
autoextend	Extends the legend to accommodate information for all layers. Should always be set to "true" with ArcMap Server.
backgroundcolor	Legend's background color using RGB values.
display	Turns legend on or off.
font	Font for title. The name is case sensitive. If font name uses "&", use "&" instead. For example, ESRI Transportation & Civic should be written as ESRI Transportation & Civic.
layerfontsize	Sets the font size of the subheading for layers that include a value

	map.
swatchheight	Height of the symbol swatch in pixels.
swatchwidth	Width of the symbol swatch in pixels.
title	Title of legend.
titlefontsize	Font size for title.
valuefontsize	For layers that include a value map, it sets the font size for value map labels. For layers with no value map, it sets the font size for the layer name.

*When parent element is **PROPERTIES when using Image Server:***

Attribute	Usage
anitaliasing	Used to make edges of labels and symbols smoother. When set to "true", antialiasing is active. Note that the time to generate a map may significantly increase when antialiasing is turned on.
autoextend	If "true", automatically extends legend vertically past size specified in height, if needed.
backgroundcolor	Legend’s background color using RGB values.
cansplit	Allows splitting of valuemap layers between columns.
cellspacing	Defines number of pixels to pad between entries.
columns	Defines number of columns in legend.
display	Turns legend on or off.
font	Font for title. The name is case sensitive. If font name uses "&", use "&" instead. For example, ESRI Transportation & Civic should be written as ESRI Transportation & Civic. For Feature Services, the font must reside on the client machine or else the system default font is used.
height	Legend height in pixels.
layerfontsize	Font size for layer labels.
reverseorder	Reverse order of layers.
splittext	Text that displays in bottom of a column if a valuemap is carried over into the next column.
swatchheight	Height of the symbol swatch in pixels.
swatchwidth	Width of the symbol swatch in pixels.
title	Title of legend.
titlefontsize	Font size for title.
transcolor	Transparency color using RGB values. White is 255,255,255.
valuefontsize	Font size for value map labels.
width	Legend width in pixels.

Examples for LEGEND:

Example 1: When in an IMAGE response and LEGEND is restricted.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <LEGEND
url="http://mymachine.domain.com/output/Iextensions_MYMACHINE4094163.jpg"
/>
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

Example 2: When in an IMAGE response with no restrictions on LEGEND.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <LEGEND file="D:\ArcIMS\output\Iextensions_MYMACHINE4094163.jpg"
url="http://mymachine.domain.com/output/Iextensions_MYMACHINE4094163.jpg"
/>
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

Example 3: GET_IMAGE request where layers with ID of "0" and "1" are removed from the legend list.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <LEGEND title="Legend" font="Arial" autoextend="true"
columns="1" width="170" height="300" backgroundcolor="255,255,0" >
          <LAYERS>
            <LAYER id="0"/>
            <LAYER id="1"/>
          </LAYERS>
        </LEGEND>
        <DRAW map="false"/>
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 4: Map configuration file where layers with ID of "0" and "1" are removed from the legend list.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP dynamic="true" >
      <PROPERTIES>
      <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      <LEGEND title="Legend" font="Arial" autoextend="true"
columns="1" width="170" height="300" backgroundcolor="255,255,0" >
        <LAYERS>
          <LAYER id="0"/>
          <LAYER id="1"/>
        </LAYERS>
      </LEGEND>
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>" />
        <SHAPEWORKSPACE name="shp_ws-2" directory="<path to USA
ESRIDATA>" />
        <SHAPEWORKSPACE name="shp_ws-3" directory="<path to CANADA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="WORLD30" visible="true" id="0">
        <DATASET name="WORLD30" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="0,153,255"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="1">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,255,153"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="United States" visible="true"
id="2">
        <DATASET name="STATES" type="polygon" workspace="shp_ws-2" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="Canada" visible="true" id="3">
        <DATASET name="province" type="polygon" workspace="shp_ws-3" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="0,153,0" />
```

```
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="World Cities" visible="true"
id="4">
        <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="102,0,102" width="8.0" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

LINE

Used in: **CONFIG** **REQUEST**
Servers: Image ArcMap
Parent elements: **OBJECT**

<LINE

coords ="x1 y1;...xn yn"
>

<HASHLINESYMBOL... /> *[Or]*
<RASTERMARKERSYMBOL... /> *[Or]*
<SIMPLELINESYMBOL... /> *[Or]*
<SIMPLEMARKERSYMBOL... /> *[Or]*
<TRUETYPEMARKERSYMBOL... /> *[Or]*

</LINE >

Bold: Attribute or child element is required.

Description:

Defines a line to be drawn on the acetate layer.

Restrictions:

- Only one symbol can be used at a time.
- Line must be continuous with no breaks.

Notes:

- For the different options of representing geometry in an acetate layer, see OBJECT.

Attribute Descriptions for LINE:

Attribute	Usage
coords	X,y coordinates representing a line. Coordinate x,y values are separated by white space, and coordinate pairs are separated by a semicolon by default. The separators can be changed by using SEPARATORS.

Examples for LINE:

Example 1: When in an acetate layer of a map configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-141.003006" miny="41.913319" maxx="-52.620281"
maxy="83.108322" name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
      <SHAPEWORKSPACE name="shp_ws-16" directory="<path to CANADA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="province" visible="true" id="0">
      <DATASET name="province" type="polygon" workspace="shp_ws-16" />
      <SIMPLERENDERER>
        <SIMPLEPOLYGONSYMBOL fillcolor="227,127,227" filltype="solid"
/>
      </SIMPLERENDERER>
      </LAYER>
      <LAYER type="acetate" name="Selectedmark" id="acetate">
        <OBJECT units="pixel">
          <LINE coords="0 0;400 0;400 13;0 13">
            <SIMPLELINESYMBOL color="0,0,0" />
        </LINE>
        </OBJECT>
     </LAYER>

    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When in an acetate layer in a GET_IMAGE request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_IMAGE>
    <PROPERTIES>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      <IMAGESIZE width="643" height="502" />
    </PROPERTIES>
    <LAYER type="acetate" name="acetate" id="acetate">
      <OBJECT units="pixel">
         <LINE coords="0 0;400 0;400 13;0 13">
            <SIMPLELINESYMBOL color="0,0,0" />
```

```
        </LINE>
      </OBJECT>
    </LAYER>
  </GET_IMAGE>
</REQUEST>
</ARCXML>
```

LOCALE

Used in: CONFIG RESPONSE Application Server RESPONSE
Parent elements: ENVIRONMENT

```
<LOCALE

    language ="string"
    country ="string"
    variant ="string"
>

    No Child Elements
</LOCALE >
```

Bold: Attribute or child element is required.

Description:

Determines the country and language locale information for an ArcIMS site.

Restrictions:

None

Notes:

None

Attribute Descriptions for LOCALE:

Attribute	Usage
country	The country code identifies differences in conventions, such as currency symbols, between countries that use the same language. Locales for a country are specified by two-letter, uppercase codes based on the ISO-3166 standard. For example, "DE" represents Germany, and "US" represents the United States.
language	The language for a locale is specified using a two-letter lowercase code based on the ISO-639 standard. For example, Spanish is "es", English is "en", and French is "fr".
variant	The variant handles variations in conventions within a language used in one country and consists of one or more underscored keywords. For example, a European country that wants to use the Euro currency symbol along with its country currency can use the variant "_EURO". If a hardware platform needs to be specified such as for Windows, the variant might be "_EURO_WIN".

Examples for LOCALE:

Example 1: When in CONFIG.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180,0" miny="-90,0" maxx="180,0" maxy="90,0"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-18" directory="<path to WORLD
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="0">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-18" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="127,227,127" filltype="solid"
/>
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When in an application server RESPONSE to GETCLIENTSERVICES.

```
<?xml version="1.0"?>
<ARCXML version="1.1">
<RESPONSE>
  <SERVICES>
    <SERVICE name="world_image" servicegroup="ImageServer1"
access="PUBLIC" type="ImageServer" desc=""  status="ENABLED" >
      <ENVIRONMENT>
        <LOCALE country="US" language="en" />
        <UIFONT name="Arial" />
      </ENVIRONMENT>
      <CLEANUP interval="20" />
    </SERVICE>
  </SERVICES>
</RESPONSE>
</ARCXML>
```

MAP

Used in: CONFIG
Parent elements: CONFIG

```
<MAP

    dynamic ="true | false"   [false]
>

    (m) <LAYER... />
    <PROPERTIES... />
    <WORKSPACES... />

</MAP >
```

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Describes map content in a map configuration file or a viewer configuration file.

Restrictions:

None

Notes:

- In order to add new map layers dynamically in a GET_IMAGE or GET_EXTRACT request, MAP must be set to dynamic.
  <MAP dynamic="true">
- For more information on configuration files, see Using Configuration Files.

Attribute Descriptions for MAP:

Attribute	Usage
dynamic	Allows users to add layers dynamically to a service without adding them to a map configuration file. Only used with requests to the Image Server and Extract Server.

Examples for MAP:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.21" miny="18.92" maxx="-66.96"
maxy="71.41" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CITIES" visible="true" id="2">
      <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLEMARKERSYMBOL type="square" width="5" />
      </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

MAPUNITS

Used in: **CONFIG** **RESPONSE**
Servers: Image Query Feature Extract ArcMap
Parent elements: **PROPERTIES**

```
<MAPUNITS

    units ="decimal_degrees | feet | meters"
>

    No Child Elements
</MAPUNITS >
```

Bold: Attribute or child element is required.

Description:

Defines data map units.

Restrictions:

None

Notes:

- In a map configuration file, MAPUNITS is optional. If it is not included, the assumption is that MAPUNITS is in decimal degrees, and no MAPUNITS information is returned in SERVICEINFO.
- If MAPUNITS is incorrect in a map configuration file, measurements and scale dependencies may not work correctly.
- If FEATURECOORDSYS is included in the map configuration file, MAPUNITS, if present, is ignored. Instead, the MAPUNITS for the coordinate system is automatically calculated. In SERVICEINFO, the correct MAPUNITS for the coordinate system is returned.
- In a request, MAPUNITS does not need to be included. The request uses the MAPUNITS in the map configuration file or the MAPUNITS from FEATURECOORDSYS.

Attribute Descriptions for MAPUNITS:

Attribute	Usage
units	Map units of the data: decimal_degrees, feet, or meters.

Examples for MAPUNITS:

Example 1: When in CONFIG.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.21" miny="18.92" maxx="-66.96"
maxy="71.41" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CITIES" visible="true" id="2">
      <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLEMARKERSYMBOL type="square" width="5" />
      </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When in a SERVICEINFO response.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-71.0718204242754" miny="42.368904975182"
maxx="-71.0475995680561" maxy="42.3869647980717" name="Initial_Extent"
/>
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="Streets"
id="1" >
        <FCLASS type="line"></FCLASS>
        <EXTENSION type="Geocode" >
          <GCSTYLE name="USAddressZ" />
```

```
        </EXTENSION>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

MARKUP

Used in: **MARKUP**
Parent elements: None

<MARKUP >
No Attributes

(m) <MARKUPLAYER... />
<WORKSPACES... />

</MARKUP >

(m): Child element can be used multiple times.

Description:

Highest level element for MARKUP; used for EditNotes.

Restrictions:

- Can only be used with Feature Service layers.

Notes:

None

Examples for MARKUP:

Example 1:

```
<?xml version="1.0"?>
<MARKUP>
  <WORKSPACES>
    <FEATURESERVERWORKSPACE name="ifs_ws-0"
url="http://mycomputer.domain.com/servlet/com.esri.esrimap.Esrimap"
service="states" />
  </WORKSPACES>
  <MARKUPLAYER layername="States" workspace="ifs_ws-0">
    <DELETEDFEATURES deletedFeatures="22">
      <FEATURE featureid="22">
      <ENVELOPE minx="-119.99807766185346" miny="34.989467399607044"
maxx="-114.03885920596963" maxy="42.00172044207277" />
      <FIELD name="AREA" precision="3" size="12" type="8">
        <FIELDVALUE valuestring="110667.293" />
      </FIELD>
      <FIELD name="STATE_NAME" precision="0" size="25" type="12">
        <FIELDVALUE valuestring="Nevada" />
      </FIELD>
      <FIELD name="STATE_FIPS" precision="0" size="2" type="12">
        <FIELDVALUE valuestring="32" />
      </FIELD>
```

```
      <FIELD name="SUB_REGION" precision="0" size="7" type="12">
        <FIELDVALUE valuestring="Mtn" />
      </FIELD>
      <FIELD name="STATE_ABBR" precision="0" size="2" type="12">
        <FIELDVALUE valuestring="NV" />
      </FIELD>
      <FIELD name="#SHAPE#" precision="0" size="0" type="-98">
        <FIELDVALUE>
          <POLYGON>
            <RING>
            <POINT x="-119.16043651343605" y="38.41182980599817" />
            <POINT x="-119.87841464065097" y="38.91441449504861" />
            ...
            <POINT x="-119.16043651343601" y="38.41182980599817" />
            <POINT x="-119.16043651343605" y="38.41182980599817" />
            </RING>
          </POLYGON>
        </FIELDVALUE>
      </FIELD>
      </FEATURE>
    </DELETEDFEATURES>
  </MARKUPLAYER>
</MARKUP>
```

MARKUPLAYER

Used in: MARKUP
Parent elements: MARKUP

<MARKUPLAYER

layername ="***string***"
workspace ="***string***"
>

<ADDEDFEATURES... />
<DELETEDFEATURES... />
<MODIFIEDFEATURES... />

</MARKUPLAYER >

Bold: Attribute or child element is required.

Description:

Identifies the layer used during an EditNotes session.

Restrictions:

None

Notes:

None

Attribute Descriptions for MARKUPLAYER:

Attribute	Usage
layername	Layer name. Must match the layer name in the ArcIMS service.
workspace	References the workspace name of the ArcIMS service.

Examples for MARKUPLAYER:

Example 1:

```
<?xml version="1.0"?>
<MARKUP>
  <WORKSPACES>
    <FEATURESERVERWORKSPACE name="ifs_ws-20"
url="http://mymachine.domain.com/servlet/com.esri.esrimap.Esrimap"
service="world" />
  </WORKSPACES>
```

```
  <MARKUPLAYER layername="CNTRY94" workspace="ifs_ws-20">
    <MODIFIEDFEATURES>
      <FEATURE featureid="1000001">
      <ENVELOPE minx="79.7" miny="-59.0" maxx="113.9" maxy="-42.4" />
      <FIELD name="AREA" precision="3" size="12" type="8">
        <FIELDVALUE valuestring="10202" />
      </FIELD>
      <FIELD name="NAME" precision="0" size="40" type="12">
        <FIELDVALUE valuestring="Atlantis" />
      </FIELD>
      <FIELD name="#SHAPE#" precision="0" size="0" type="-98">
        <FIELDVALUE>
          <POLYGON>
            <RING>
            <POINT x="85.61944739721136" y="-42.43367913036056" />
            <POINT x="113.93068023991125" y="-46.33867676383642" />
            <POINT x="80.25007565118213" y="-55.12492143915705" />
            <POINT x="85.61944739721136" y="-42.43367913036056" />
            </RING>
          </POLYGON>
        </FIELDVALUE>
      </FIELD>
      </FEATURE>
    </MODIFIEDFEATURES>
  </MARKUPLAYER>
</MARKUP>
```

METADATA

Used in: RESPONSE
Servers: Metadata (Browse)
Parent elements: RESPONSE

```
<METADATA

    numresults ="integer"
    startresult ="string"
    total ="integer"
>

    <COLLECTION_INFO... />  [Or]
    <CONTENT_INFO... />  [Or]
    (m) <METADATA_DATASET... />  [Or]
    <USER... />  [Or]
    (m) <UUID... />  [Or]

</METADATA >
```

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Parent element used with all metadata responses.

Restrictions:

- Only one child element is used per response.
- Attributes for METADATA are returned only for SEARCH_METADATA requests.

Notes:

None

Attribute Descriptions for METADATA:

Attribute	Usage
numresults	Number of datasets returned in the response.
startresult	Returns *startresult* attribute value used in request.
total	Total number of datasets found in database for the given search criteria.

Examples for METADATA:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <METADATA numresults="1" startresult="1" total="1" >
      <METADATA_DATASET name="Addresses" owner="homer"
userid="{6754F64F-931F-11D5-99E7-000086460FAA}"
gnd="http://mycomputer.esri.com/output/myfile.gnd"
onlink="http://out.there.com/website/mywebsite"
server="http://out.there.com" service="AddMatch" servicetype="image"
content="image" url="http://mycomputer.esri.com/output/myfile.xml"
children="false" siblings="false" private="false" folder="false"
refcount="1" >
        <ENVELOPE minx="-98.100000039814" miny="42.9000000090804"
maxx="-95.900000067288" maxy="44.100000039814" />
      </METADATA_DATASET>
    </METADATA>
  </RESPONSE>
</ARCXML>
```

METADATA_ACTION

Used in: RESPONSE
Servers: Metadata
Parent elements: RESPONSE

<METADATA_ACTION >
No Attributes

No Child Elements
</METADATA_ACTION >

Description:

Response from metadata requests that require confirmation that the request was successful.

Restrictions:

None

Notes:

- METADATA_ACTION is returned in response to the following requests:
  - ADD_RELEVANCE_FEEDBACK
  - CHANGE_METADATA_ACCESS
  - DELETE_METADATA
  - DELETE_METADATA_RELATIONSHIP
  - PUT_METADATA
  - PUT_METADATA_RELATIONSHIP
  - PUT_METADATA_SEMANTIC
  - PUT_USER
  - RENAME_METADATA
  - RESET
- Either "OK" or an error message is returned between the opening and closing METADATA_ACTION elements.

Examples for METADATA_ACTION:

Example 1:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <METADATA_ACTION>OK</METADATA_ACTION>
  </RESPONSE>
</ARCXML>
```

METADATA_CHILD

Used in: REQUEST
Servers: Metadata (Publish)
Parent elements: DELETE_METADATA_RELATIONSHIP
PUT_METADATA_RELATIONSHIP

```
<METADATA_CHILD

    docid ="string"
>

    No Child Elements
</METADATA_CHILD >
```

Bold: Attribute or child element is required.

Description:

Identifies a child document that has a relationship to a source or parent metadata document.

Restrictions:

None

Notes:

None

Attribute Descriptions for METADATA_CHILD:

Attribute	Usage
docid	String used to uniquely identify a document. The client used to publish the metadata is responsible for creating the document ID. This ID is automatically assigned when using ArcCatalog. If another client is used, GET_UUID can be used to request a valid ID. The format for an ID is the following: {HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH} where H is a hexadecimal digit (0-9,a-f,A-F). The ID is limited to 38 characters.

Examples for METADATA_CHILD:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <PUBLISH_METADATA>
      <PUT_METADATA_RELATIONSHIP>
        <METADATA_SOURCE docid="{C64D8F38-82B4-11D5-99C2-000086460FA0}"
/>
        <METADATA_CHILD docid="{F99A8C01-76B3-62E5-12G4-
056786460CA0}" />
      </PUT_METADATA_RELATIONSHIP>
    </PUBLISH_METADATA>
  </REQUEST>
</ARCXML>
```

METADATA_CONFIG

Used in: CONFIG
Parent elements: CONFIG

<METADATA_CONFIG >
No Attributes

<ADMIN_TABLE... />
<METADATA_CONTENT... />
<OUTPUT... />
<TABLE_NAME... />

</METADATA_CONFIG >

Description:

Main element for defining a metadata configuration file.

Restrictions:

None

Notes:

- This element is not available from the ArcIMS Author interface.

Examples for METADATA_CONFIG:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <METADATA_CONFIG>
      <WORKSPACES>
        <SDEWORKSPACE name="unique_name" server="server_name"
instance="port:5151" database="optional_database_name" user="user_name"
password="user_password" />
      </WORKSPACES>
      <METADATA_CONTENT validate="true" index_words="automatic" />
      <TABLE_NAME prefix="imsmetadata" />
    </METADATA_CONFIG>
  </CONFIG>
</ARCXML>
```

METADATA_CONTENT

Used in: CONFIG
Parent elements: METADATA_CONFIG

```
<METADATA_CONTENT

    index_words ="automatic | manual"   [automatic]
    minwordlength ="0 - NNN"   [3]
    private ="true | false"   [false]
    validate ="true | false"   [true]
>

    No Child Elements
</METADATA_CONTENT >
```

Description:

Specifies whether attribute validation is required for the service.

Restrictions:

None

Notes:

- The following items of content are central to the operations of searching and viewing search results in the Metadata Explorer. It is recommended that all published documents contain these items: Title, Publisher, Content type, Data theme, and Extent. These items are required if *validate* is "true" in METADATA_CONTENT.
- This element is not available from the ArcIMS Author interface.

Attribute Descriptions for METADATA_CONTENT:

Attribute	Usage
index_words	In order to do a text search on a document, the document must be indexed. By default, a document is indexed when it is published through ArcCatalog. Set *index_words* to "automatic" for automatic indexing. Indexing can also take place offline using a standalone indexer. Set *index_words* to "manual" for manual indexing offline. Note that if a document is not indexed, only spatial searches are valid and no text searches can be made.
minwordlength	Sets the minimum word length for indexing. A minimum word length of "0" means every character is indexed separately. The recommended minimum word length is "3" except for languages such as Japanese, Chinese, and Thai.

private	Determines whether document is available to all viewers or is a private document. By default, documents are public.
validate	Validation is set to "true" by default. If set to "false", clients do not validate which items of content are available.

Examples for METADATA_CONTENT:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <METADATA_CONFIG>
      <WORKSPACES>
        <SDEWORKSPACE name="unique_name" server="server_name"
instance="port:5151" database="optional_database_name" user="user_name"
password="user_password" />
      </WORKSPACES>
      <METADATA_CONTENT validate="true" index_words="automatic" />
      <TABLE_NAME prefix="imsmetadata" />
    </METADATA_CONFIG>
  </CONFIG>
</ARCXML>
```

METADATA_DATASET

Used in: RESPONSE
Servers: Metadata (Browse)
Parent elements: METADATA

```
<METADATA_DATASET

    children="true | false"
    docid ="string"
    folder="true | false"
    index_status ="indexed | not indexed | indexing error"
    name ="string"
    owner="string"
    private ="true | false"
    refcount ="integer"
    siblings ="true | false"
    timestamp ="string"
    content ="liveData | downloadableData | offlineData | staticMapImage | document |
application | geographicService | clearinghouse | unknown"
    gnd ="string"
    onlink ="string"
    server ="string"
    service ="string"
    servicetype ="metadata | image | feature | wms"
    thumbnail ="string"
    updated ="string"
    url ="string"
>

    <ENVELOPE... />

</METADATA_DATASET >
```

Bold: Attribute or child element is required.

Description:

Describes a metadata document in the metadata repository.

Restrictions:

None

Notes:

- See SEARCH_METADATA, GET_METADATA_DOCUMENT, and GET_ROOT_DATASET for request.
- Column names added through RESPONSE_COLUMN *columnname* become "custom attributes" in METADATA_DATASET. For example, if you include RESPONSE_COLUMN *columnname="service_running"*, then in addition to the extisting attributes in METADATA_DATASET, a new attribute named *service_running* will always be returned as part of METADATA_DATASET. Please note the following:
    - If you are using a DTD, you must update METADATA_DATASET to account for these "custom attributes".
    - You cannot use a column name that replicates any existing attribute in METADATA_DATASET.

Attribute Descriptions for METADATA_DATASET:

Attribute	Usage
children	Set to "true" if the dataset has children.
content	Metadata document content type.
docid	String used to uniquely identify a document. The client used to publish the metadata is responsible for creating the document ID. This ID is automatically assigned when using ArcCatalog. If another client is used, GET_UUID can be used to request a valid ID. The format for an ID is the following: {HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH} where H is a hexadecimal digit (0-9,a-f,A-F). The ID is limited to 38 characters.
folder	Set to "true" if the document represents a folder.
gnd	URL of a GND file on the server pointing to an ArcIMS service referenced by the metadata document.
index_status	Determines whether a document is indexed for text searches. If an error has occurred while indexing, "indexing error" is returned.
name	Name that identifies the dataset corresponding to the document.
onlink	A string identifying the location of the dataset corresponding to the document.
owner	Name identifying the owner of the metadata document.
private	Set to "false" if the document is viewable by all users. Set to "true" if viewable only by the owner.
refcount	Number of times the document is referenced in the hierarchy.
server	The URL of the Web server containing data or metadata associated with the document, for example, http://mymachine.domain.com.
service	The name of the ArcIMS service containing data or metadata associated with the document.

servicetype	Type of ArcIMS service.
siblings	Set to "true" if the dataset has siblings.
thumbnail	URL of the thumbnail image.
timestamp	Date and time of last insertion or update into the metadata repository.
updated	Last date metadata record was updated. Format is YYYY-MM-DD hh:mm:ss.
url	URL of the XML document being retrieved.

Examples for METADATA_DATASET:

Example 1: Includes the "custom attribute" called Approved

```
<?xml version="1.0" encoding="UTF8" ?>
<ARCXML version="1.1">
  <RESPONSE>
    <METADATA numresults="1" startresult="0" total="1">
      <METADATA_DATASET name="metadata" owner="author"
docid="{CD4AEFEF-896E-45EC-9A7E-EEBA823370C5}" content="unknown"
url="http://mymachine.domain.com/output/OracleMetadata_P375_T437_D18.xml"
children="true" siblings="false" private="false" folder="true"
index_status="indexed" refcount="1" updated="2002-02-11 14:51:14"
Approved="Y" >
        <ENVELOPE minx="-141.001235918609" miny="41.3912889520516"
maxx="-71.2933350698463" maxy="68.6637039277661" />
      </METADATA_DATASET>
    </METADATA>
  </RESPONSE>
</ARCXML>
```

METADATA_SIBLING

Used in: REQUEST
Servers: Metadata (Publish)
Parent elements: DELETE_METADATA_RELATIONSHIP
PUT_METADATA_RELATIONSHIP

```
<METADATA_SIBLING

    docid ="string"
>

    No Child Elements
</METADATA_SIBLING >
```

Bold: Attribute or child element is required.

Description:

Identifies a sibling, or related, document to a source metadata document.

Restrictions:

None

Notes:

- A sibling document is a related document to the source document. When selected as a sibling, the document is listed with the parent document under a link called "related records".

Attribute Descriptions for METADATA_SIBLING:

Attribute	Usage
docid	String used to uniquely identify a document. The client used to publish the metadata is responsible for creating the document ID. This ID is automatically assigned when using ArcCatalog. If another client is used, GET_UUID can be used to request a valid ID. The format for an ID is the following: {HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH} where H is a hexadecimal digit (0-9,a-f,A-F). The ID is limited to 38 characters.

Examples for METADATA_SIBLING:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <PUBLISH_METADATA>
      <PUT_METADATA_RELATIONSHIP>
        <METADATA_SOURCE docid="{C64D8F38-82B4-11D5-99C2-000086460FA0}"
/>
        <METADATA_SIBLING docid="{1F7DDF21-BC01-4C20-8AA5-
243B33ED0B1E}" />
      </PUT_METADATA_RELATIONSHIP>
    </PUBLISH_METADATA>
  </REQUEST>
</ARCXML>
```

METADATA_SOURCE

Used in: REQUEST
Servers: Metadata (Publish)
Parent elements: DELETE_METADATA_RELATIONSHIP
PUT_METADATA_RELATIONSHIP

```
<METADATA_SOURCE

    docid ="string"
>

    No Child Elements
</METADATA_SOURCE >
```

Bold: Attribute or child element is required.

Description:

Identifies the source or parent metadata document.

Restrictions:

None

Notes:

None

Attribute Descriptions for METADATA_SOURCE:

Attribute	Usage
docid	String used to uniquely identify a document. The client used to publish the metadata is responsible for creating the document ID. This ID is automatically assigned when using ArcCatalog. If another client is used, GET_UUID can be used to request a valid ID. The format for an ID is the following: {HHHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH} where H is a hexadecimal digit (0-9,a-f,A-F). The ID is limited to 38 characters.

Examples for METADATA_SOURCE:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <PUBLISH_METADATA>
      <PUT_METADATA_RELATIONSHIP>
        <METADATA_SOURCE docid="{C64D8F38-82B4-11D5-99C2-
000086460FA0}" />
        <METADATA_CHILD docid="{F99A8C01-76B3-62E5-12G4-056786460CA0}"
/>
      </PUT_METADATA_RELATIONSHIP>
    </PUBLISH_METADATA>
  </REQUEST>
</ARCXML>
```

MODIFIEDFEATURES

Used in: MARKUP
Parent elements: MARKUPLAYER

<MODIFIEDFEATURES >
No Attributes

(m) <FEATURE... />

</MODIFIEDFEATURES >

(m): Child element can be used multiple times.

Description:

Defines features modified in a layer during an EditNotes session.

Restrictions:

None

Notes:

None

Examples for MODIFIEDFEATURES:

Example 1:

```
<?xml version="1.0"?>
<MARKUP>
  <WORKSPACES>
    <FEATURESERVERWORKSPACE name="ifs_ws-20"
url="http://mymachine.domain.com/servlet/com.esri.esrimap.Esrimap"
service="world" />
  </WORKSPACES>
  <MARKUPLAYER layername="CNTRY94" workspace="ifs_ws-20">
    <MODIFIEDFEATURES>
      <FEATURE featureid="1000001">
      <ENVELOPE minx="79.7" miny="-59.0" maxx="113.9" maxy="-42.4" />
      <FIELD name="AREA" precision="3" size="12" type="8">
        <FIELDVALUE valuestring="10202" />
      </FIELD>
      <FIELD name="NAME" precision="0" size="40" type="12">
        <FIELDVALUE valuestring="Atlantis" />
      </FIELD>
      <FIELD name="#SHAPE#" precision="0" size="0" type="-98">
        <FIELDVALUE>
          <POLYGON>
            <RING>
            <POINT x="85.61944739721136" y="-42.43367913036056" />
```

```
            <POINT x="113.93068023991125" y="-46.33867676383642" />
            <POINT x="80.25007565118213" y="-55.12492143915705" />
            <POINT x="85.61944739721136" y="-42.43367913036056" />
            </RING>
          </POLYGON>
        </FIELDVALUE>
      </FIELD>
      </FEATURE>
    </MODIFIEDFEATURES>
  </MARKUPLAYER>
</MARKUP>
```

MOIMSWORKSPACE

Used in: CONFIG
Parent elements: WORKSPACES

```
<MOIMSWORKSPACE

    name ="string"
    service ="string"
    url ="string"
>

    No Child Elements
</MOIMSWORKSPACE >
```

Bold: Attribute or child element is required.

Description:

Specifies a workspace for an ArcExplorer-enabled MapObjects IMS application.

Note: this element has been deprecated in ArcIMS 4.0.

Restrictions:

- Limited to ArcExplorer-enabled MapObjects IMS 2.0 applications. Not valid with MapObjects 2.1 or higher.
- Can only be used in viewer configuration files. It cannot be used in a map configuration file.

Notes:

- For more information on using MapObjects IMS 2.0 applications with ArcIMS, see ArcIMS Online.

Attribute Descriptions for MOIMSWORKSPACE:

Attribute	Usage
name	Workspace name. Must be unique among all data sources.
service	ArcIMS service name.
url	Point to the location of esrimap.dll or esrimapn.dll.

Examples for MOIMSWORKSPACE:

Example 1: When in a viewer configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-117.0" miny="12.2" maxx="-86.7" maxy="35.0"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <MOIMSWORKSPACE name="mo_ws-4"
url="http://mycomputer.domain.com/scripts/esrimap.dll" service="Mexico"
/>
      </WORKSPACES>
      <LAYER type="image" name="arcims2:tahoe" visible="true" id="0">
        <DATASET name="arcims2:tahoe" type="image" workspace="mo_ws-4"
/>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

MULTIPOINT

Used in: **CONFIG** **REQUEST** **RESPONSE** **MARKUP**
Servers: Image Query Feature Extract ArcMap
Parent elements: **FEATURE** **FIELDVALUE** **OBJECT** **SPATIALFILTER**

<MULTIPOINT >
No Attributes

<COORDS... /> *[Or]*
(m) <POINT... /> *[Or]*

</MULTIPOINT >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Represents data for a point feature value.

Restrictions:

- Either COORDS or POINT must be used as a child element.

Notes:

- For the different options of representing geometry in an acetate layer, see OBJECT.

Examples for MULTIPOINT:

Example 1: When in CONFIG and parent element is SPATIALFILTER.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
    <PROPERTIES>
      <ENVELOPE minx="-178.21502685546875" miny="18.924781799316406"
maxx="-66.9698486328125" maxy="71.40664672851562" name="Initial_Extent"
/>
      <MAPUNITS units="decimal_degrees" />
    </PROPERTIES>
    <WORKSPACES>
```

```
      <SHAPEWORKSPACE name="shp_ws-4" directory="<path to USA
ESRIDATA>" />
    </WORKSPACES>
    <LAYER type="featureclass" name="STATES" visible="true" id="0">
      <DATASET name="STATES" type="polygon" workspace="shp_ws-4" />
        <SPATIALQUERY>
          <SPATIALFILTER relation="area_intersection">
            <MULTIPOINT>
              <COORDS> -113.10 37.99;-110.99 43.88;-95.99 44.88
</COORDS>
            </MULTIPOINT>
          </SPATIALFILTER>
        </SPATIALQUERY>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="127,227,27"/>
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When in a GET_IMAGE request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_IMAGE>
    <PROPERTIES>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      <IMAGESIZE width="643" height="502" />
    </PROPERTIES>
    <LAYER type="featureclass" name="select layer" visible="true"
id="300">
      <DATASET fromlayer="Countries" />
      <SPATIALQUERY>
        <SPATIALFILTER relation="area_intersection">
           <MULTIPOINT>
             <POINT x="100.33758678275998" y="62.81971775984563" />
          </MULTIPOINT>
        </SPATIALFILTER>
      </SPATIALQUERY>
      <SIMPLERENDERER>
        <SIMPLEPOLYGONSYMBOL fillcolor="0,255,0" filltype="cross"
fillinterval="3" />
      </SIMPLERENDERER>
    </LAYER>
  </GET_IMAGE>
</REQUEST>
</ARCXML>
```

Example 3: When using MULTIPOINT in an acetate layer.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
```

```
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <IMAGESIZE width="800" height="600" />
      </PROPERTIES>
      <LAYER type="acetate" name="acetate" id="acetate">
        <OBJECT units="database">
          <SIMPLEMARKERSYMBOL color="0,0,0" width="20" />
          <MULTIPOINT>
             <POINT x="-120.1079549837513" y="40.99815142335011" />
             <POINT x="-110.99468788980437" y="30.88488432940315" />
             <POINT x="10.13210788980437" y="30.82488950980315" />
          </MULTIPOINT>
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 4: When in a FEATURES response if the GET_FEATURES request includes outputmode="xml" and geometry="true".

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <FEATURES>
      <FEATURE>
        <FIELDS NAME="Los Angeles" COUNTRY="US" POPULATION="9763600"
CAPITAL="N" #SHAPE#="[Geometry]" #ID#="534" />
        <MULTIPOINT>
          <POINT x="-118.25" y="34" />
        </MULTIPOINT>
      </FEATURE>
      <FEATURECOUNT count="1" hasmore="false" />
    </FEATURES>
  </RESPONSE>
</ARCXML>
```

Example 5: When in MARKUP.

```
<?xml version="1.0"?>
<MARKUP>
  <WORKSPACES>
    <FEATURESERVERWORKSPACE name="ifs_ws-1"
url="http://mymachine.domain.com/servlet/com.esri.esrimap.Esrimap"
service="usa" />
  </WORKSPACES>
  <MARKUPLAYER layername="Cities" workspace="ifs_ws-1">
    <ADDEDFEATURES>
      <FEATURE featureid="1000000">
      <ENVELOPE minx="-118.7325331443268" miny="40.698835264575465"
maxx="-118.7325331443268" maxy="40.698835264575465" />
      <FIELD name="NAME" precision="0" size="40" type="12" />
      <FIELD name="COUNTRY" precision="0" size="12" type="12" />
```

```
      <FIELD name="POPULATION" precision="0" size="11" type="4" />
      <FIELD name="CAPITAL" precision="0" size="1" type="12" />
      <FIELD name="#SHAPE#" precision="0" size="0" type="-98">
        <FIELDVALUE>
          <MULTIPOINT>
            <POINT x="-118.7325331443268" y="40.698835264575465" />
          </MULTIPOINT>
        </FIELDVALUE>
      </FIELD>
      </FEATURE>
    </ADDEDFEATURES>
  </MARKUPLAYER>
</MARKUP>
```

NORTHARROW

Used in: **CONFIG** **REQUEST**
Servers: Image ArcMap
Parent elements: **OBJECT**

<NORTHARROW

When using ArcMap Server:
coords **="*double*"**
type **="1 - 8"**
angle ="0.0 - 360.0" **[0]**
size ="*integer*" **[30]**

When using Image Server:
coords **="*double*"**
type **="1 - 8"**
angle ="0.0 - 360.0" **[0]**
antialiasing ="true | false" **[false]**
outline ="0,0,0 - 255,255,255"
overlap ="true | false" **[true]**
shadow ="0,0,0 - 255,255,255"
size ="*integer*" **[30]**
transparency ="0.0 - 1.0" **[1.0]**
>

No Child Elements
</NORTHARROW >

Bold: Attribute or child element is required.

Description:

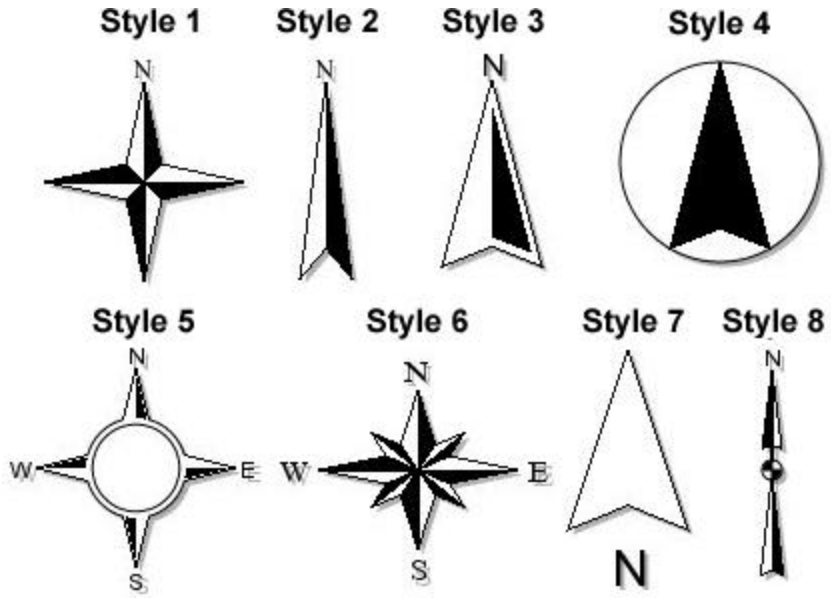
Places a north arrow on the acetate layer of the map.

Restrictions:

- NORTHARROW, as an object in an acetate layer, can only be used with Image and ArcMap Image Services in an HTML Viewer.

Notes:

- The following north arrow types are available:



- Custom north arrows can be used in an acetate layer. Use RASTERMARKERSYMBOL or TRUETYPEMARKERSYMBOL inside of POINT. NORTHARROW does not support custom arrows.

Attribute Descriptions for NORTHARROW:

Attribute	Usage
angle	Angle of the north arrow in degrees. 0 degrees points to the north, and values increase moving clockwise.
antialiasing	Used to make edges of labels and symbols smoother. When set to "true", antialiasing is active. Note that the time to generate a map may significantly increase when antialiasing is turned on.
coords	North arrow placement location. Coordinate pair is separated by white space by default. The separator can be changed by using SEPARATORS. If using pixel coordinates, "0 0" is in the lower left corner of the map viewer area.
outline	Outline color using RGB values.
overlap	Determines if labels can overlap this symbol. When "true", labels can overlap. When "false", labels will not overlap the symbol. If labels are not drawing as expected, check if *overlap* is set to "false" for this symbol or any other symbol in the ArcIMS service.
shadow	Shadow color using RGB values.
size	Arrow size in pixels.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.
type	Value representing north arrow type.

Examples for NORTHARROW:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_IMAGE>
    <PROPERTIES>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      <IMAGESIZE width="643" height="502" />
    </PROPERTIES>
    <LAYER type="acetate" name="acetate" id="acetate">
      <OBJECT units="pixel">
        <NORTHARROW type="4" size="15" coords="20 30"
shadow="32,32,32" angle="0" antialiasing="True" overlap="False"/>
      </OBJECT>
    </LAYER>
  </GET_IMAGE>
</REQUEST>
</ARCXML>
```

OBJECT

Used in: **CONFIG** **REQUEST**
Servers: Image ArcMap
Parent elements: **LAYER**

<OBJECT

When using ArcMap Server:
units ="database | pixel"

When using Image Server:
units ="database | pixel"
lower ="*string*" **[1:1]**
upper ="*string*" **[1:infinity]**
>

***For lines, option 1 with Image and ArcMap Image Services is to use* LINE:**
<LINE... />
<COORDSYS... />

***For lines, option 2 with ArcMap Image Services is to use* POLYLINE *and one symbol*:**
<POLYLINE... />
<HASHLINESYMBOL... /> *[Or]*
<SIMPLELINESYMBOL... /> *[Or]*
<COORDSYS... />

***For lines, option 2 with Image Services is to use* POLYLINE *and one symbol*:**
<POLYLINE... />
<HASHLINESYMBOL... /> *[Or]*
<RASTERMARKERSYMBOL... /> *[Or]*
<SIMPLELINESYMBOL... /> *[Or]*
<SIMPLEMARKERSYMBOL... /> *[Or]*
<TRUETYPEMARKERSYMBOL... /> *[Or]*
<COORDSYS... />

***For points, option 1 with Image and ArcMap Image Services is to use* POINT:**
<POINT... />
<COORDSYS... />

***For points, option 2 with Image and ArcMap Image Services is to use* MULTIPOINT *and one symbol*:**
<MULTIPOINT... />
<RASTERMARKERSYMBOL... /> *[Or]*

<SIMPLEMARKERSYMBOL... /> *[Or]*
<TRUETYPEMARKERSYMBOL... /> *[Or]*
<COORDSYS... />

***For polygons, option 1 with Image and ArcMap Image Services is to use* POLYGON. RING and HOLE are not permitted:**
<POLYGON... />
<COORDSYS... />

***For polygons, option 2 with ArcMap Image Services is to use* POLYGON *and one symbol*. RING and HOLE are permitted:**
<POLYGON... />
<GRADIENTFILLSYMBOL... /> *[Or]*
<RASTERFILLSYMBOL... /> *[Or]*
<SIMPLEPOLYGONSYMBOL... /> *[Or]*
<COORDSYS... />

***For polygons, option 2 with Image Services is to use* POLYGON *and one symbol*. RING and HOLE are permitted:**
<POLYGON... />
<GRADIENTFILLSYMBOL... /> *[Or]*
<HASHLINESYMBOL... /> *[Or]*
<RASTERFILLSYMBOL... /> *[Or]*
<RASTERMARKERSYMBOL... /> *[Or]*
<SIMPLELINESYMBOL... /> *[Or]*
<SIMPLEMARKERSYMBOL... /> *[Or]*
<SIMPLEPOLYGONSYMBOL... /> *[Or]*
<TRUETYPEMARKERSYMBOL... /> *[Or]*
<COORDSYS... />

Other OBJECT child elements with Image and ArcMap Image Services:
<NORTHARROW... /> *[Or]*
<SCALEBAR... /> *[Or]*
<TEXT... /> *[Or]*
<COORDSYS... />

</OBJECT >

Bold: Attribute or child element is required.

Description:

Defines an object to be placed on an acetate layer.

Restrictions:

- When drawing lines, point, or polygons on an acetate layer, either Option 1 or Option 2 can be used, but not both. If Option 2 is selected, one symbol is also required. NORTHARROW, SCALEBAR, and TEXT have only one option for specifying a location.
- Acetate layers can only be used with Image and ArcMap Image Services in HTML Viewers.
- The attributes *upper* and *lower* are valid only with Image Server.
- When an object is projected on an ArcMap Image Service, points and line and polygon vertices will be placed properly when *units="database"*. However, line segments and polygon edges will be straight rather than projected.

Notes:

- In acetate layers, two options are available for defining the location of a point, line, or polygon. In Option 1 for points, POINT includes the geometry, and one symbol element must be included as a child element of POINT. In Option 2, MULTIPOINT includes the geometry. The symbol is included as a child element of OBJECT.

Option 1 for Points	Option 2 for Points
<OBJECT units="pixel"> <POINT coords="0 0"> <SYMBOL> <POINT> </OBJECT>	<OBJECT units="pixel" > <SYMBOL> <MULTIPOINT> <POINT x="0" y="0" /> </MULTIPOINT> </OBJECT>

In Option 1 for lines, LINE includes the geometry, and one symbol element must be included as a child element of LINE. In Option 2, POLYLINE, PATH, and either POINT or COORDS include the geometry. The symbol is included as a child element of OBJECT. Note that with Option 2, multiple PATHs can be included within one POLYLINE.

Option 1 for Lines	Option 2 for Lines
<OBJECT units="pixel"> <LINE coords="0 0;400 0;400 13;0 13"> <SYMBOL /> </LINE> </OBJECT>	<OBJECT units="database"> <SYMBOL /> <POLYLINE> <PATH> <COORDS>0 0;400 0;400 13;0 13</COORDS> </PATH> </POLYLINE> </OBJECT>

In Option 1 for polygons, POLYGON includes the geometry, and one symbol element must be included as a child element of POLYGON. In Option 2, POLYGON, RING, HOLE, and either POINT or COORDS include the geometry. The symbol is included as a child element of OBJECT. Note that with Option 2, multiple RINGS and HOLEs can be included within one POLGYON.

Option 1 for Polygons	Option 2 for Polygons
<OBJECT units="pixel"> <POLYGON coords="10 10;400 10;400 20;10 20;10 10"> <SYMBOL /> </POLYGON> </OBJECT>	<OBJECT units="database"> <SYMBOL /> <POLYGON> <RING> <POINT x="83.15" y="38.07" /> <POINT x="111.09" y="-4.70" /> <POINT x="155.10" y="-10.38" /> <POINT x="139.10" y="66.38" /> <POINT x="83.15" y="38.07" /> <HOLE> <COORDS>100.15 20.07;103.09 30.70;106.10 30.38;100.15 20.07</COORDS> </HOLE> </RING> </POLYGON> </OBJECT>

- In order to project an acetate layer, COORDSYS must be included as a child element of OBJECT. COORDSYS is valid only when OBJECT *units="database"*. The COORDSYS *id* should specify the current coordinate system of the x,y coordinates listed within the OBJECT.
- Scales can be set in ArcXML using a relative scale or by calculating the number of map units per pixel. A relative scale represents the scale in a ratio such as 1:24000. In this example, 1 meter equals 24000 meters, or 1 inch equals 24000 inches. When using relative scale, always use a colon (:) between the two values.

  Map units per pixel refers to the number of meters, feet, or decimal degrees represented by one pixel in a map. To convert from a relative scale to map units per pixel, the size of a pixel must first be calculated. The formula for finding the number of meters in a pixel is 0.0254 / dpi. The value 0.0254 is the number of meters in an inch, and dpi is the dpi set in the ArcIMS service or request. If no dpi is set in the service or request, the dpi is assumed to be 96. As an example of pixel size, if the dpi is 96, the pixel size is 0.0254 / 96 or 0.000265 m. To convert from a relative scale to map units per pixel:

  1. If the scale is in **meters**. To calculate the number of meters per pixel, take the relative scale and multiply by 0.000265. For example, if the relative scale is 1:24000, then the number of meters per pixel is 24000 * 0.000265, or 6.36 meters.
  2. If the scale is in **feet**. Do the calculation for meters (#1). Multiply the result by 3.28 (the number of feet in a meter). For example, if the number of meters per pixel is 6.36, the number of feet is 6.36 * 3.28, or 20.86 feet.
  3. If the scale is in **decimal degrees**. For these calculations, the earth is assumed to be an exact circle with a circumference of 40030.174 km. One degree is 111.195 km (40030.174/360 degrees), or 111195 meters. To calculate the number of degrees, first do the calculation for meters (#1).

Next, divide the result by 111195. For example, if the number of meters per pixel is 6.36, the number of degrees is 6.36 / 111195, or 0.0000571968.

Attribute Descriptions for OBJECT:

Attribute	Usage
lower	Minimum scale to display an object using a relative scale such as 1:24000. Scale can also be calculated as the number of map units per pixel.
units	Determines how coordinates for the object are specified. Coordinates can be specified two ways: • Database. Refers to positioning an object using x,y coordinates in the coordinate system of the ArcIMS service or request. For example, if the service is in Robinson, then the coordinates for the object should also be in Robinson. If the coordinates for the object are different from the coordinate system used in the service or request, then COORDSYS should be used. <OBJECT...> <COORDSYS.../> ... </OBJECT> • Pixel. Refers to positioning an object using x,y coordinates in pixels. The pixels along the left edge of the map frame have an x-coordinate of zero. The pixels along the bottom edge have a y-coordinate of zero.
upper	Maximum scale to display an object using a relative scale such as 1:24000. Scale can also be calculated as the number of map units per pixel.

Examples for OBJECT:

Example 1: When in CONFIG.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-141.003006" miny="41.913319" maxx="-52.620281"
maxy="83.108322" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-16" directory="<path to CANADA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="province" visible="true" id="0">
        <DATASET name="province" type="polygon" workspace="shp_ws-16"
/>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="227,127,227" filltype="solid"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="acetate" name="Selectedmark" id="acetate">
        <OBJECT units="pixel">
          <NORTHARROW type="4" size="15" coords="20 30"
shadow="32,32,32" angle="0" antialiasing="True" overlap="False"/>
        </OBJECT>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When in a GET_IMAGE request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      <IMAGESIZE width="643" height="502" />
      </PROPERTIES>
      <LAYER type="acetate" name="acetate" id="acetate">
      <OBJECT units="pixel">
          <TEXT coords="100 100" label="You are here">
            <TEXTMARKERSYMBOL font="Arial" />
        </TEXT>
```

```
      </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 3: When using POINT for one acetate layer and MULTIPOINT for a second acetate layer.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <IMAGESIZE width="800" height="600" />
      </PROPERTIES>
      <LAYER type="acetate" name="acetate" id="acetate">
        <OBJECT units="pixel">
          <POINT coords="100 100" >
            <SIMPLEMARKERSYMBOL color="0,0,0" width="20" />
          </POINT>
        </OBJECT>
      </LAYER>
      <LAYER type="acetate" name="acetate" id="acetate">
        <OBJECT units="database">
          <SIMPLEMARKERSYMBOL color="0,0,0" width="20" />
          <MULTIPOINT>
             <POINT x="-120.1079549837513" y="40.99815142335011" />
             <POINT x="-110.99468788980437" y="30.88488432940315" />
             <POINT x="10.13210788980437" y="30.82488950980315" />
          </MULTIPOINT>
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 4: When using POLYLINE for one acetate layer and LINE for a second acetate layer.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <IMAGESIZE width="800" height="600" />
      </PROPERTIES>
      <LAYER type="acetate" name="Selectedmark" id="acetate">
        <OBJECT units="database">
          <SIMPLELINESYMBOL color="0,0,0" width="6" />
```

```
          <POLYLINE>
            <PATH>
              <POINT x="-2.1079549837513" y="19.99815142335011" />
              <POINT x="28.99468788980437" y="15.88488432940315" />
              <POINT x="55.99468788980437" y="35.88488432940315" />
            </PATH>
          </POLYLINE>
        </OBJECT>
      </LAYER>
      <LAYER type="acetate" name="Selectedmark" id="acetate1">
        <OBJECT units="pixel">
          <LINE coords="0 0;400 0;400 13;0 13">
            <SIMPLELINESYMBOL color="0,255,0" width="6" />
          </LINE>
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 5: When using POLYGON (Option 1) for one acetate layer and POLYGON, RING, and HOLE (Option 2) for a second acetate layer.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <IMAGESIZE width="800" height="600" />
      </PROPERTIES>
      <LAYER type="acetate" name="acetate" id="acetate">
        <OBJECT units="pixel">
          <POLYGON coords="10 10;400 10;400 20;10 20;10 10">
            <SIMPLEPOLYGONSYMBOL fillcolor="255,255,0" />
          </POLYGON>
        </OBJECT>
      </LAYER>
      <LAYER type="acetate" name="acetate1" id="acetate1">
        <OBJECT units="database">
          <SIMPLEPOLYGONSYMBOL fillcolor="0,255,0" />
          <POLYGON>
            <RING>
              <POINT x="83.15605550814075" y="38.07185101549165" />
              <POINT x="111.09942196116728" y="-4.70645066589869" />
              <POINT x="155.1079549837513" y="-10.38915084069517" />
              <POINT x="139.1079549837513" y="66.38915084069517" />
              <POINT x="83.15605550814075" y="38.07185101549165" />
              <HOLE>
                <POINT x="100.15605550814075" y="20.07185101549165" />
                <POINT x="103.09942196116728" y="30.70645066589869" />
                <POINT x="106.1079549837513" y="30.38915084069517" />
                <POINT x="100.15605550814075" y="20.07185101549165" />
              </HOLE>
```

```
            </RING>
          </POLYGON>
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

OTHER

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Feature
Parent elements: **VALUEMAPLABELRENDERER** **VALUEMAPRENDERER**

<OTHER

label ="*string*"
>

*When parent element is **VALUEMAPLABELRENDERER:***
<CALLOUTMARKERSYMBOL... /> *[Or]*
<CHARTSYMBOL... /> *[Or]*
<RASTERSHIELDSYMBOL... /> *[Or]*
<SHIELDSYMBOL... /> *[Or]*
<TEXTSYMBOL... /> *[Or]*

*When parent element is **VALUEMAPRENDERER:***
<GRADIENTFILLSYMBOL... /> *[Or]*
<HASHLINESYMBOL... /> *[Or]*
<RASTERFILLSYMBOL... /> *[Or]*
<RASTERMARKERSYMBOL... /> *[Or]*
<SIMPLELINESYMBOL... /> *[Or]*
<SIMPLEMARKERSYMBOL... /> *[Or]*
<SIMPLEPOLYGONSYMBOL... /> *[Or]*
<TRUETYPEMARKERSYMBOL... /> *[Or]*

</OTHER >

Bold: Attribute or child element is required.

Description:

Used with VALUEMAPRENDERER and VALUEMAPLABELRENDERER as the default for rendering symbols that do not meet the criteria for any RANGE or EXACT values.

Restrictions:

- One renderer child element is required with OTHER.
- Not valid with ArcMap Server.

Notes:

- If DATASET layer type is point, then only point symbols can be used. If type is line, then line and point symbols can be used. If type is polygon, then polygon, line, and point symbols can be used.
- OTHER can be used in the same value map with EXACT or RANGE but is not required.
- For more information on using renderers, see Using ArcXML Renderers.

Attribute Descriptions for OTHER:

Attribute	Usage
label	Label for legend.

Examples for OTHER:

Example 1: When in CONFIG or REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="Zoning" visible="true" id="1"
minscale="1:20000" maxscale="1:30000">
        <DATASET name="ZONING" type="polygon" workspace="shp_ws-48" />
        <VALUEMAPRENDERER lookupfield="ZONE">
        <EXACT value="RA-MH" label="RA-MH">
          <SIMPLEPOLYGONSYMBOL fillcolor="206,171,88" filltype="solid"
boundarycolor="206,171,88" />
        </EXACT>
        <EXACT value="O&amp;amp;R-2" label="O&amp;amp;R-2">
          <SIMPLEPOLYGONSYMBOL fillcolor="216,11,254" filltype="solid"
boundarycolor="216,11,254" />
        </EXACT>
        <OTHER>
          <SIMPLEPOLYGONSYMBOL fillcolor="128,128,128"
boundarycolor="0,0,0" />
        </OTHER>
        </VALUEMAPRENDERER>
```

```
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When using a label with OTHER (available for image MapServices in an HTML viewer only).

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-178.215027" miny="18.924782" maxx="-66.969849"
maxy="71.406647" name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
      <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="STATES" visible="true" id="0">
      <DATASET name="STATES" type="polygon" workspace="shp_ws-0" />
      <VALUEMAPRENDERER lookupfield="STATE_ABBR">
        <EXACT value="NV;NE;ND;NC;NY;NH;NJ;NM" label="States with N">
          <SIMPLEPOLYGONSYMBOL fillcolor="27,27,127" filltype="solid"
/>
        </EXACT>
        <EXACT value="MT;ME;MA;MD;MS;MO;MI;MN" label="States with M">
          <SIMPLEPOLYGONSYMBOL fillcolor="227,227,227" filltype="solid"
/>
        </EXACT>
        <OTHER label="All Other States">
          <SIMPLEPOLYGONSYMBOL fillcolor="127,27,27" filltype="solid"
/>
        </OTHER>
      </VALUEMAPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

OUTPUT

Used in: CONFIG REQUEST RESPONSE
Servers: Image Extract ArcMap
Parent elements: EXTRACT IMAGE LAYOUT METADATA_CONFIG PROPERTIES

<OUTPUT

*When parent element is **IMAGE, EXTRACT, LAYOUT:***
file ="***string***"
url ="***string***"
height ="*integer*"
width ="*integer*"

*When parent element is **METADATA_CONFIG and using the Metadata Server:***
baseurl ="*string*"
path ="*string*"

*When parent element is **PROPERTIES in GET_EXTRACT:***
baseurl ="*string*"
name ="*string*"
path ="*string*"
url ="*string*"

*When parent element is **PROPERTIES in GET_IMAGE:***
baseurl ="*string*"
legendbaseurl ="*string*"
legendname ="*string*"
legendpath ="*string*"
legendurl ="*string*"
name ="*string*"
path ="*string*"
type ="gif | jpg | png | png8" **[jpg]**
url ="*string*"

*When parent element is **PROPERTIES in GET_LAYOUT:***
baseurl ="*string*"
name ="*string*"
path ="*string*"
type ="ai | bmp | emf | eps | jpg | pdf | png8 | png24" **[jpg]**
url ="*string*"
>

No Child Elements
</OUTPUT >

Bold: Attribute or child element is required.

Description:

Defines pathname and URL for output map images, legend images, and ZIP files containing extracted shapefiles.

Restrictions:

- GIF format is not supported as an output format with the ArcMap Server. GIF format can be used with the Image Server.
- When using an ArcIMS HTML Viewer, ArcIMS Java Viewer, ArcExplorer 4, or any other browser using the ArcIMS Servlet Connector, the OUTPUT element is restricted.
    - In IMAGE and EXTRACT responses, the attribute *file* is not returned.
    - In GET_IMAGE and GET_EXTRACT requests, OUTPUT is ignored.
    - In map configuration files, OUTPUT is ignored and is not included in a SERVICEINFO response.

  These restrictions can be lifted by setting the properties *spatialServer.AllowRequestOutput* and *spatialServer.AllowResponsePath* to true in esrimap_prop. This property file is found in the servlet engine directory. For more information on the location of esrimap_prop and its properties, see the ArcIMS Help.

  These restrictions apply only when the ArcIMS Servlet Connector is used. They do not apply to the ActiveX, ColdFusion, or Java Connectors.

- OUTPUT works with paired attributes. If one of the attributes is used, its pair is also required. The attribute pairs are listed in the table below.

Attribute	Paired Attribute	Filename Assignment
path	baseurl	ArcIMS assigns random filename.
name	url	User assigns a filename.
legendpath	legendbaseurl	ArcIMS assigns random filename.
legendname	legendurl	User assigns a filename.

- When using GET_EXTRACT, the filename extension is restricted to *.zip. When using GET_IMAGE, filename extensions are restricted to *.jpg, *.png, and *.gif. When using GET_LAYOUT, filename extensions are restricted to *.ai, *.bmp, *.emf, *.eps, *.jpg, *.pdf, and *.png.

Notes:

- When not restricted, and OUTPUT is used in a map configuration file, the output file location overrides the location assigned when the service was started. If OUTPUT is used in a request, the file output location in the request overrides both OUTPUT in the map configuration file and the location assigned when the service was started.
- If OUTPUT is used, the output files are not automatically deleted by ArcIMS Tasker. In order for the files to be deleted, the *taskfile* property must be set in tasker.properties. For information on setting this property, see the ArcIMS Help.
- For more information on using OUTPUT, see Using GET_IMAGE and IMAGE with Image Services, Using GET_IMAGE and IMAGE with ArcMap Image Services and Using GET_EXTRACT and EXTRACT.

Attribute Descriptions for OUTPUT:

*When parent element is **IMAGE, EXTRACT, LAYOUT:***

Attribute	Usage
file	Full pathname and filename for location of map image generated by the ArcIMS Spatial Server. UNC pathnames are valid (\\myComputer\arcims\output).
height	If *autoresize* is used in a GET_IMAGE request, and the requested map is larger than the image memory limit, then the generated map is resized to fit within the image memory limit. In this case, the height of the output image is included in the IMAGE response.
url	URL used by client to retrieve map image.
width	If *autoresize* is used in a GET_IMAGE request, and the requested map is larger than the image memory limit, then the generated map is resized to fit within the image memory limit. In this case, the width of the image is included in the IMAGE response.

*When parent element is **METADATA_CONFIG and using the Metadata Server:***

Attribute	Usage
baseurl	Paired with *path*. URL of output directory if default filename is generated by ArcIMS. Do not include a filename.
path	Paired with *baseurl*. Directory to output file generated by the Metadata Server. Do not include the filename. UNC pathnames are valid (\\myComputer\arcims\output).

*When parent element is **PROPERTIES in GET_EXTRACT:***

Attribute	Usage
baseurl	Paired with *path*. URL of output directory if default filename is generated by ArcIMS. Do not include a filename.
name	Paired with *url*. User assigns an output ZIP filename. Only filenames with a *.zip extension are valid. Use full pathname along with the

	filename. The filename must match the filename used in *url*. UNC pathnames are valid (\\myComputer\arcims\output\myfile.zip).
path	Paired with *baseurl*. Directory to output ZIP file generated by Extract Server. Do not include the filename. UNC pathnames are valid (\\myComputer\arcims\output).
url	Paired with *name*. URL of output ZIP file. Include filename as part of URL. The filename must match the filename used in *name*.

*When parent element is **PROPERTIES in GET_IMAGE:***

Attribute	Usage
baseurl	Paired with *path*. URL of output directory if default filename is generated by ArcIMS. Do not include a filename.
legendbaseurl	Paired with *legendpath*. URL of output directory if default legend filename is generated by ArcIMS. Do not include a filename.
legendname	Paired with *legendurl*. User assigns an output legend filename. Use full pathname along with the filename. The filename must match the filename used in *legendurl*. UNC pathnames are valid (\\myComputer\arcims\output\mylegend.jpg).
legendpath	Paired with *legendbaseurl*. Directory to output legend file generated by Image Server. Do not include the filename. UNC pathnames are valid (\\myComputer\arcims\output).
legendurl	Paired with *legendname*. URL of output legend file. Include filename as part of URL. The filename must match the filename used in *legendname*.
name	Paired with *url*. User assigns an output filename. Use full pathname along with the filename. Only filenames with a *.jpg, *.png, or *.gif extension are valid. The filename must match the filename used in *url*. UNC pathnames are valid (\\myComputer\arcims\output\myfile.jpg).
path	Paired with *baseurl*. Directory to output file generated by the Image Server. Do not include the filename. UNC pathnames are valid (\\myComputer\arcims\output).
type	Output image file type. Note that the GIF format is not supported with ArcMap Server.
url	Paired with *name*. URL of output file. Include filename as part of URL. The filename must match the filename used in *name*.

*When parent element is **PROPERTIES in GET_LAYOUT:***

Attribute	Usage
baseurl	Paired with *path*. URL of output directory if default filename is generated by ArcIMS. Do not include a filename.
name	Paired with *url*. User assigns an output filename. Use full pathname along with the filename. Only filenames with a *.ai, *.bmp, *.emf,

	*.eps, *.jpg, *.pdf, or *.png extension are valid. The filename must match the filename used in *url*. UNC pathnames are valid (\\myComputer\arcims\output\myfile.jpg).
path	Paired with *baseurl*. Directory to output file generated by the ArcMap Server. Do not include the filename. UNC pathnames are valid (\\myComputer\arcims\output).
type	Output file type. Can be ai, bmp, emf, eps, jpg, pdf, png8 (8 bit), png24 (24 bit).
url	Paired with *name*. URL of output file. Include filename as part of URL. The filename must match the filename used in *name*.

Examples for OUTPUT:

Example 1: When in a RESPONSE with OUTPUT restricted.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-80.00000000" miny="-56.00000000"
maxx="80.00000000" maxy="56.00000000" />
      <OUTPUT
url="http://mymachine.domain.com/maps/WorldMap_MYMACHINE16114829.jpg"
/>
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

Example 2: When in a RESPONSE with no restrictions on OUTPUT.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-80.00000000" miny="-56.00000000"
maxx="80.00000000" maxy="56.00000000" />
      <OUTPUT file="F:\WorldMap_MYMACHINE16114829.jpg"
url="http://mymachine.domain.com/maps/WorldMap_MYMACHINE16114829.jpg"
/>
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

Example 3: In a RESPONSE when autoresize is used in GET_IMAGE, and the image has been resized.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
```

```
    <IMAGE>
      <ENVELOPE minx="-180" miny="-135" maxx="180" maxy="135" />
      <OUTPUT file="c:\arcims\output\world_MYMACHINE8236420.png"
url="http://mymachine.domain.com/output/world_MYMACHINE8236420.png"
width="591" height="443"/>
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

Example 4: When in a REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      <IMAGESIZE width="643" height="502" />
      <OUTPUT name="c:\arcims\myfile.jpg"
url="http://mymachine.domain.com/myfile.jpg" />
      </PROPERTIES>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 5: Map configuration file that includes OUTPUT information for both the image and legend. Name/url attribute pairs are used.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP dynamic="true" >
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
        <OUTPUT legendname="c:\arcims\website\legend.jpg"
legendurl="http://mymachine.domain.com/website/legend.jpg"
name="c:\arcims\website\map.jpg"
url="http://mymachine.domain.com/website/map.jpg"  />
        <LEGEND title="Legend" font="Arial" columns="1" width="170"
height="300" backgroundcolor="255,255,0" >
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="CITIES" visible="true" id="4">
        <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
        <SIMPLERENDERER>
```

```
          <SIMPLEMARKERSYMBOL color="102,0,102" width="8.0" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

OUTPUTFIELD

Used in: **CONFIG** **RESPONSE**
Servers: Extract
Parent elements: **OUTPUTFILE**

<OUTPUTFIELD

alias ="*string*"
name ="*string*"
>

No Child Elements
</OUTPUTFIELD >

Bold: Attribute or child element is required.

Description:

Sets up alias names for fields in a shapefile generated using the Extract Server.

Restrictions:

None

Notes:

- For more information on using the Extract Server, see Using GET_EXTRACT and EXTRACT.

Attribute Descriptions for OUTPUTFIELD:

Attribute	Usage
alias	Alias name for field in the database. Shapefiles use DBF files to store attribute data, and fields in a DBF are limited to 10 characters. Since other databases often allow more than 10 characters, it is possible to end up with two fields in the DBF file with the same name. ArcIMS will not allow shapefiles with duplicate field names to be used. In this scenario, an alias name should be used.
name	Name of field in the database.

Examples for OUTPUTFIELD:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP dynamic="true">
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-2" directory="<path to USA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="STATES" visible="true"
id="States">
        <DATASET name="STATES" type="polygon" workspace="shp_ws-2" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="CITIES" visible="true" id="35">
        <DATASET name="CITIES" type="point" workspace="shp_ws-2" />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="102,0,102" width="8.0" />
        </SIMPLERENDERER>
        <SPATIALQUERY where="POP1990 &gt; 100000" subfields="CITY_NAME
STATE_NAME POP1990 MALES FEMALES"  />
        <EXTENSION type="extract" >
        <EXTRACTPARAMS clip="true" >
          <OUTPUTFILE file="us_cities" >
            <OUTPUTFIELD name="CITY_NAME" alias="City" />
            <OUTPUTFIELD name="STATE_NAME" alias="State" />
            <OUTPUTFIELD name="POP1990" alias="Population" />
            <OUTPUTFIELD name="MALES" alias="Male_pop" />
            <OUTPUTFIELD name="FEMALES" alias="Female_pop" />
          </OUTPUTFILE>
        </EXTRACTPARAMS>
        </EXTENSION>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

OUTPUTFILE

Used in: **CONFIG** **RESPONSE**
Servers: Extract
Parent elements: **EXTRACTPARAMS**

<OUTPUTFILE

file ="*string*"
>

(m) <OUTPUTFIELD... />

</OUTPUTFILE >

(m): Child element can be used multiple times.

Description:

Sets up an alias name for a shapefile generated using the Extract Server.

Restrictions:

- Output file names are limited to English characters.

Notes:

- By default, the resulting shapefile names in a GET_EXTRACT request are based on the LAYER *id* in a map configuration file. Characters are limited to those that are also valid for a filename. OUTPUTFILE allows for an alternative name to be used for the shapefile name.
- For more information on using the Extract Server, see Using GET_EXTRACT and EXTRACT.

Attribute Descriptions for OUTPUTFILE:

Attribute	Usage
file	An alias name for the shapefile that is generated.

Examples for OUTPUTFILE:

Example 1: OUTPUTFILE "us_cities" is used to identify "CITIES" layer in a GET_EXTRACT request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
```

```
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP dynamic="true">
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-2" directory="<path to USA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="STATES" visible="true"
id="States">
        <DATASET name="STATES" type="polygon" workspace="shp_ws-2" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="CITIES" visible="true" id="35">
        <DATASET name="CITIES" type="point" workspace="shp_ws-2" />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="102,0,102" width="8.0" />
        </SIMPLERENDERER>
        <SPATIALQUERY where="POP1990 &gt; 100000" subfields="CITY_NAME
STATE_NAME POP1990 MALES FEMALES"  />
        <EXTENSION type="extract" >
        <EXTRACTPARAMS clip="true" >
          <OUTPUTFILE file="us_cities" >
            <OUTPUTFIELD name="CITY_NAME" alias="City" />
            <OUTPUTFIELD name="STATE_NAME" alias="State" />
            <OUTPUTFIELD name="POP1990" alias="Population" />
            <OUTPUTFIELD name="MALES" alias="Male_pop" />
            <OUTPUTFIELD name="FEMALES" alias="Female_pop" />
          </OUTPUTFILE>
        </EXTRACTPARAMS>
        </EXTENSION>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

OVERVIEWMAP

Used in: CONFIG
Parent elements: CONFIG

<OVERVIEWMAP

backgroundcolor ="0,0,0 - 255,255,2555" **[255,255,255]**
framefillcolor ="0,0,0 - 255,255,255" **[255,0,0]**
frameoutlinecolor ="0,0,0 - 255,255,255" **[255,0,0]**
zoomfactor ="*double*"
>

(m) <LAYERDEF... />

</OVERVIEWMAP >

(m): Child element can be used multiple times.

Description:

Defines an overview map for ArcIMS Java clients.

Restrictions:

- Used only for ArcExplorer 4 and ArcIMS Java Viewers.

Notes:

- If no layers are added to the overview map using LAYERDEF, the overview map is opened with no layers in it.
- OVERVIEWMAP is added to viewer configuration files and saved in ArcExplorer 4 or the ArcIMS Java Standard Viewer if the overview map is open.

Attribute Descriptions for OVERVIEWMAP:

Attribute	Usage
backgroundcolor	Background color using RGB values.
framefillcolor	Frame fill color using RGB values.
frameoutlinecolor	Frame outline color using RGB values.
zoomfactor	Ratio of overview map extent to main map extent. Small positive values should be used.

Examples for OVERVIEWMAP:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-116.016078" miny="36.252371" maxx="-100.855887"
maxy="46.622450" name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
      <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="STATES" visible="true" id="0">
      <DATASET name="STATES" type="polygon" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLEPOLYGONSYMBOL fillcolor="255,0,0" />
      </SIMPLERENDERER>
      </LAYER>
    </MAP>
    <OVERVIEWMAP backgroundcolor="255,255,255"
framefillcolor="255,0,0" frameoutlinecolor="255,0,0" zoomfactor="4.0">
      <LAYERDEF name="STATES" />
    </OVERVIEWMAP>
  </CONFIG>
</ARCXML>
```

PARTITION

Used in: CONFIG
Parent elements: DATASET

```
<PARTITION

    name ="string"
>

    <ENVELOPE... />

</PARTITION >
```

Bold: Attribute or child element is required.

Description:

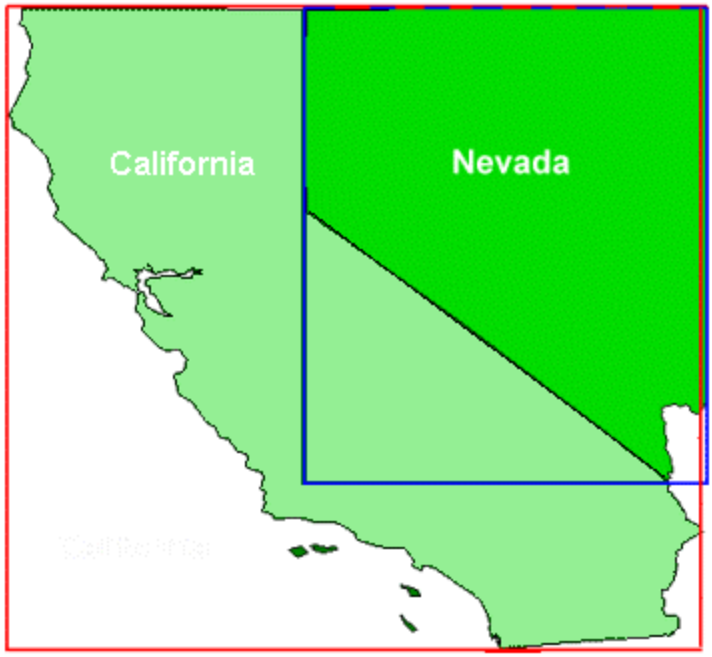
A method of grouping multiple layers and treating them as one big layer.

Restrictions:

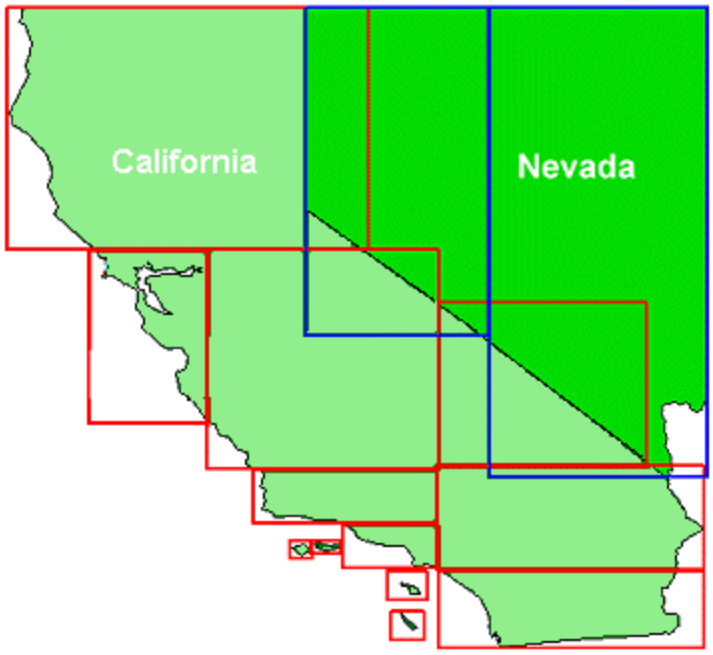
- Partitioned layers can be used only on ArcSDE layers and not with shapefiles.
- Partitioned layers cannot be geocoded.
- Partitioned layers cannot have related tables.
- Partitioned layers do not support stored queries.
- Partitioned layers work only with vector ArcSDE layers and cannot be used with raster layers.

Notes:

- PARTITION is a method of grouping multiple layers and treating them as one layer. Each piece of the partitioned layer is a tile segment that must include an ENVELOPE. The partitioned layers can be a traditional tile where no envelopes overlap, or the partitioned layers can also have overlapping envelopes. For example, if data from two states is used as partitioned layers, the envelopes will likely overlap. In the case of California and Nevada, the extents of both states overlap each other as shown in the figure below. During a query, if a point lies within both extents, both partitioned layers are queried.



A layer can be partitioned multiple times. In the next example showing California and Nevada, the states are divided into a series of smaller partitions. Within each state, no partitions overlap, although they could. There is still some overlap between the Nevada and California partitions, but the overlap area between the two states is much smaller. The advantage of using smaller partitions is that smaller areas are searched during a query and the overlapping extents are smaller.



Each PARTITION envelope is an index to the layer. If data in a layer is not included in one of the PARTITION envelopes, depending on the requested extent,

the data may or may not be retrieved. If the requested extent is completely outside any of the PARTITION envelopes, no data is returned. If the requested extent includes part of a PARTITION envelope, then the data is returned.

- All layers described within a PARTITION must be present in the ArcSDE database. If one reference is incorrect, the entire layer will not display.
- DATASET *name* can be an arbitrary name. The actual ArcSDE layer names must be specified in PARTITION *name*.

Attribute Descriptions for PARTITION:

Attribute	Usage
name	Layer name for partition. Specifies ArcSDE layer. Multiple ArcSDE layers with the same name are permitted. In the PARTITION example, CALIFORNIA.STREET02 is used twice. The ENVELOPE is different for each PARTITION.

Examples for PARTITION:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-124.0" miny="32.0" maxx="-114.0" maxy="42.0"/>
        <BACKGROUND color="255,255,204"/>
      </PROPERTIES>
      <WORKSPACES>
        <SDEWORKSPACE name="sde-1" server="CALIFORNIA"
instance="port:5150" user="street_data" password="AFXOR"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="Streets" visible="true" id="1">
        <DATASET name="CALIFORNIA.STREET" type="line" workspace="sde-
1">
          <PARTITION name="CALIFORNIA.STREET01">
              <ENVELOPE minx="-124.0" miny="36.0" maxx="-116.0"
maxy="42.0"/>
          </PARTITION>
          <PARTITION name="CALIFORNIA.STREET02">
              <ENVELOPE minx="-122.0" miny="33.0" maxx="-114.0"
maxy="36.0"/>
          </PARTITION>
          <PARTITION name="CALIFORNIA.STREET02">
              <ENVELOPE minx="-117.0" miny="32.0" maxx="-113.0"
maxy="33.0"/>
```

```
          </PARTITION>
        </DATASET>
        <SIMPLERENDERER>
           <SIMPLELINESYMBOL type="solid" width="3" color="102,102,102"
/>
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

PATH

Used in: CONFIG REQUEST RESPONSE MARKUP
Servers: Image Query Feature Extract ArcMap
Parent elements: POLYLINE

<PATH >
No Attributes

<COORDS... /> *[Or]*
(m) <POINT... /> *[Or]*

</PATH >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Represents data for a polyline feature.

Restrictions:

- Either COORDS or POINT is required.

Notes:

- For the different options of representing geometry in an acetate layer, see OBJECT.

Examples for PATH:

Example 1: When in SPATIALFILTER in CONFIG or REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <IMAGESIZE width="643" height="502" />
      </PROPERTIES>
      <LAYER type="featureclass" name="select layer" visible="true"
id="Selected">
        <DATASET fromlayer="1" />
        <SPATIALQUERY>
          <SPATIALFILTER relation="area_intersection">
            <POLYLINE>
              <PATH>
                <COORDS>-2.10 19.99;28.99 15.88;55.99 35.88 </COORDS>
              </PATH>
            </POLYLINE>
```

```
          </SPATIALFILTER>
        </SPATIALQUERY>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="0,255,0" filltype="cross"
fillinterval="3" />
        </SIMPLERENDERER>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 2: When using PATH in an acetate layer.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <IMAGESIZE width="800" height="600" />
      </PROPERTIES>
      <LAYER type="acetate" name="Selectedmark" id="acetate">
        <OBJECT units="database">
          <SIMPLELINESYMBOL color="0,0,0" width="6" />
          <POLYLINE>
            <PATH>
              <POINT x="-2.1079549837513" y="19.99815142335011" />
              <POINT x="28.99468788980437" y="15.88488432940315" />
              <POINT x="55.99468788980437" y="35.88488432940315" />
            </PATH>
          </POLYLINE>
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 3: When in a FEATURES response.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
<RESPONSE>
  <FEATURES>
    <FEATURE>
      <FIELDS CUST_ID="4" NAME="Customer  4"  #SHAPE#="[Geometry]"
#ID#="3" />
      <POLYLINE>
        <PATH>
          <POINT x="-128.1079549837513" y="81.99815142335011" />
          <POINT x="-123.99468788980437" y="77.88488432940315" />
          <POINT x="-123.99468788980437" y="77.88488432940315" />
      </PATH>
      </POLYLINE>
    </FEATURE>
```

```
    <FEATURECOUNT count="1" hasmore="false" />
  </FEATURES>
</RESPONSE>
</ARCXML>
```

POINT

Used in: **CONFIG** **REQUEST** **RESPONSE** **MARKUP**
Servers: Image Query Feature Extract Geocode ArcMap
Parent elements: **FIELDVALUE** **HOLE** **MULTIPOINT** **OBJECT** **PATH** **RING**

<POINT

*When parent element is **FIELDVALUE, MULTIPOINT, PATH, RING, HOLE:***
x ="***double***"
y ="***double***"

*When parent element is **OBJECT:***
coords ="***double***"
>

*When parent element is **FIELDVALUE, MULTIPOINT, PATH, RING, HOLE:***
No Child Elements

*When parent element is **OBJECT:***
<**RASTERMARKERSYMBOL...** /> *[Or]*
<**SIMPLEMARKERSYMBOL...** /> *[Or]*
<**TRUETYPEMARKERSYMBOL...** /> *[Or]*

</POINT >

Bold: Attribute or child element is required.

Description:

Defines a point based on an x,y coordinate.

Restrictions:

- For acetate layers, only one symbol can be used at a time.

Notes:

- For the different options of representing geometry in an acetate layer, see OBJECT.

Attribute Descriptions for POINT:

*When parent element is **FIELDVALUE, MULTIPOINT, PATH, RING, HOLE:***

Attribute	Usage
x	x-coordinate for point.
y	y-coordinate for point.

*When parent element is **OBJECT:***

Attribute	Usage
coords	Point placement location. Coordinate pair is separated by white space by default. The separator can be changed by using SEPARATORS. If using pixel coordinates, "0 0" is in the lower left corner of the map viewer area.

Examples for POINT:

Example 1: When in an acetate layer in CONFIG.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-141.003006" miny="41.913319" maxx="-52.620281"
maxy="83.108322" name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-16" directory="<path to CANADA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="province" visible="true" id="0">
      <DATASET name="province" type="polygon" workspace="shp_ws-16" />
      <SIMPLERENDERER>
        <SIMPLEPOLYGONSYMBOL fillcolor="227,127,227" filltype="solid"
/>
      </SIMPLERENDERER>
      </LAYER>
      <LAYER type="acetate" name="Selectedmark" id="acetate" >
      <OBJECT units="pixel">
          <POINT coords="250 300">
            <SIMPLEMARKERSYMBOL color="0,0,0" />
          </POINT>
      </OBJECT>
      </LAYER>
    </MAP>
  </CONFIG>
```

```
</ARCXML>
```

Example 2: When in an acetate layer in a GET_IMAGE request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_IMAGE>
    <PROPERTIES>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      <IMAGESIZE width="643" height="502" />
    </PROPERTIES>
    <LAYER type="acetate" name="acetate" id="acetate">
      <OBJECT units="pixel">
         <POINT coords="100 100" >
            <SIMPLEMARKERSYMBOL color="0,0,0" />
         </POINT>
      </OBJECT>
    </LAYER>
  </GET_IMAGE>
</REQUEST>
</ARCXML>
```

Example 3: When using POINT and PATH in an acetate layer.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <IMAGESIZE width="800" height="600" />
      </PROPERTIES>
      <LAYER type="acetate" name="Selectedmark" id="acetate">
        <OBJECT units="database">
          <SIMPLELINESYMBOL color="0,0,0" width="6" />
          <POLYLINE>
            <PATH>
              <POINT x="-2.1079549837513" y="19.99815142335011" />
              <POINT x="28.99468788980437" y="15.88488432940315" />
              <POINT x="55.99468788980437" y="35.88488432940315" />
            </PATH>
          </POLYLINE>
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 4: When using POINT for one acetate layer and MULTIPOINT for a second acetate layer.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <IMAGESIZE width="800" height="600" />
      </PROPERTIES>
      <LAYER type="acetate" name="acetate" id="acetate">
        <OBJECT units="pixel">
          <POINT coords="100 100" >
            <SIMPLEMARKERSYMBOL color="0,0,0" width="20" />
          </POINT>
        </OBJECT>
      </LAYER>
      <LAYER type="acetate" name="acetate" id="acetate">
        <OBJECT units="database">
          <SIMPLEMARKERSYMBOL color="0,0,0" width="20" />
          <MULTIPOINT>
             <POINT x="-120.1079549837513" y="40.99815142335011" />
             <POINT x="-110.99468788980437" y="30.88488432940315" />
             <POINT x="10.13210788980437" y="30.82488950980315" />
          </MULTIPOINT>
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 5: When in SPATIALFILTER in CONFIG or REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<CONFIG>
  <MAP>
    <PROPERTIES>
      <ENVELOPE minx="-180.0" miny="-152.0" maxx="180.0" maxy="153.0"
/>
      <MAPUNITS units="decimal_degrees" />
    </PROPERTIES>
    <WORKSPACES>
      <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
    </WORKSPACES>
    <LAYER type="Featureclass" name="Cities" visible="true" id="0">
      <DATASET name="cities" type="point" workspace="shp_ws-0" />
      <SPATIALQUERY>
      <SPATIALFILTER relation="area_intersection">
         <MULTIPOINT>
             <POINT x="-128.1079549837513" y="81.99815142335011" />
             <POINT x="-123.99468788980437" y="77.88488432940315" />
             <POINT x="-123.99468788980437" y="77.88488432940315" />
         </MULTIPOINT>
      </SPATIALFILTER>
```

```
      </SPATIALQUERY>
      <SIMPLERENDERER>
        <TRUETYPEMARKERSYMBOL font="ESRI Cartography" character="252"
/>
      </SIMPLERENDERER>
   </LAYER>
</MAP>
</CONFIG>
</ARCXML>
```

Example 6: When in a FEATURES response.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
<RESPONSE>
  <FEATURES>
    <FEATURE>
      <FIELDS CUST_ID="4" NAME="Customer  4"  #SHAPE#="[Geometry]"
#ID#="3" />
      <POLYLINE>
        <PATH>
          <POINT x="-128.1079549837513" y="81.99815142335011" />
          <POINT x="-123.99468788980437" y="77.88488432940315" />
          <POINT x="-123.99468788980437" y="77.88488432940315" />
      </PATH>
      </POLYLINE>
    </FEATURE>
    <FEATURECOUNT count="1" hasmore="false" />
  </FEATURES>
</RESPONSE>
</ARCXML>
```

Example 7: When in MARKUP.

```
<?xml version="1.0"?>
<MARKUP>
  <WORKSPACES>
    <FEATURESERVERWORKSPACE name="ifs_ws-20"
url="http://mymachine.domain.com/servlet/com.esri.esrimap.Esrimap"
service="world" />
  </WORKSPACES>
  <MARKUPLAYER layername="CNTRY94" workspace="ifs_ws-20">
    <MODIFIEDFEATURES>
      <FEATURE featureid="1000001">
      <ENVELOPE minx="79.7" miny="-59.0" maxx="113.9" maxy="-42.4" />
      <FIELD name="AREA" precision="3" size="12" type="8">
        <FIELDVALUE valuestring="10202" />
      </FIELD>
      <FIELD name="NAME" precision="0" size="40" type="12">
        <FIELDVALUE valuestring="Atlantis" />
      </FIELD>
      <FIELD name="#SHAPE#" precision="0" size="0" type="-98">
        <FIELDVALUE>
          <POLYGON>
            <RING>
```

```
            <POINT x="85.61944739721136" y="-42.43367913036056" />
            <POINT x="113.93068023991125" y="-46.33867676383642" />
            <POINT x="80.25007565118213" y="-55.12492143915705" />
            <POINT x="85.61944739721136" y="-42.43367913036056" />
            </RING>
          </POLYGON>
        </FIELDVALUE>
      </FIELD>
      </FEATURE>
    </MODIFIEDFEATURES>
  </MARKUPLAYER>
</MARKUP>
```

POLYGON

Used in: **CONFIG** **REQUEST** **RESPONSE** **MARKUP**
Servers: Image Query Feature Extract ArcMap
Parent elements: **FEATURE** **FIELDVALUE** **OBJECT** **SPATIALFILTER**

<POLYGON

*When parent element is **FIELDVALUE, FEATURE, SPATIALFILTER, OBJECT:***
No Attributes

*When parent element is **OBJECT:***
coords ="x1 y1;...xn yn"
>

*When parent element is **FIELDVALUE, FEATURE, SPATIALFILTER, OBJECT:***
(m) <RING... />

*When parent element is **OBJECT:***
<GRADIENTFILLSYMBOL... /> *[Or]*
<HASHLINESYMBOL... /> *[Or]*
<RASTERFILLSYMBOL... /> *[Or]*
<RASTERMARKERSYMBOL... /> *[Or]*
<SIMPLELINESYMBOL... /> *[Or]*
<SIMPLEMARKERSYMBOL... /> *[Or]*
<SIMPLEPOLYGONSYMBOL... /> *[Or]*
<TRUETYPEMARKERSYMBOL... /> *[Or]*

</POLYGON >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Renders a polygon on the acetate layer using the OBJECT element or defines a polygon using the FIELDVALUE, FEATURE, or SPATIALFILTER elements.

Restrictions:

- In an acetate layer, one symbol element is required, and only one can be used at a time.
- The polygon must be closed.

Notes:

- For the different options of representing geometry in an acetate layer, see OBJECT.

Attribute Descriptions for POLYGON:

*When parent element is **FIELDVALUE, FEATURE, SPATIALFILTER, OBJECT:***
No Attributes

*When parent element is **OBJECT:***

Attribute	Usage
Cords	X,y coordinates representing a line. Coordinate x,y values are separated by white space, and coordinate pairs are separated by a semicolon by default. The separators can be changed by using SEPARATORS.

Examples for POLYGON:

Example 1: When using POLYGON in acetate layers.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <IMAGESIZE width="800" height="600" />
      </PROPERTIES>
      <LAYER type="acetate" name="acetate" id="acetate">
        <OBJECT units="pixel">
          <POLYGON coords="10 10;400 10;400 20;10 20;10 10">
            <SIMPLEPOLYGONSYMBOL fillcolor="255,255,0" />
          </POLYGON>
        </OBJECT>
      </LAYER>
      <LAYER type="acetate" name="acetate1" id="acetate1">
        <OBJECT units="database">
          <SIMPLEPOLYGONSYMBOL fillcolor="0,255,0" />
          <POLYGON>
            <RING>
              <POINT x="83.15605550814075" y="38.07185101549165" />
              <POINT x="111.09942196116728" y="-4.70645066589869" />
              <POINT x="155.1079549837513" y="-10.38915084069517" />
              <POINT x="139.1079549837513" y="66.38915084069517" />
              <POINT x="83.15605550814075" y="38.07185101549165" />
              <HOLE>
                <POINT x="100.15605550814075" y="20.07185101549165" />
                <POINT x="103.09942196116728" y="30.70645066589869" />
                <POINT x="106.1079549837513" y="30.38915084069517" />
                <POINT x="100.15605550814075" y="20.07185101549165" />
              </HOLE>
            </RING>
          </POLYGON>
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
```

```
  </REQUEST>
</ARCXML>
```

Example 2: When in an acetate layer in CONFIG.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-141.003006" miny="41.913319" maxx="-52.620281"
maxy="83.108322" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-16" directory="<path to CANADA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="province" visible="true" id="0">
        <DATASET name="province" type="polygon" workspace="shp_ws-16"
/>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="227,127,227" filltype="solid"
/>
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="acetate" name="Selectedmark" id="acetate">
        <OBJECT units="pixel">
           <POLYGON coords="0 0;400 0;400 13;0 13;0 0">
            <SIMPLEPOLYGONSYMBOL fillcolor="0,0,0" />
          </POLYGON>
        </OBJECT>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 3: When in an acetate layer in a GET_IMAGE request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_IMAGE>
    <PROPERTIES>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      <IMAGESIZE width="643" height="502" />
    </PROPERTIES>
    <LAYER type="acetate" name="acetate" id="acetate">
      <OBJECT units="pixel">
      <POLYGON coords="0 0;400 0;400 13;0 13;0 0">
```

```
      <SIMPLEPOLYGONSYMBOL fillcolor="255,255,0" />
      </POLYGON>
      </OBJECT>
    </LAYER>
  </GET_IMAGE>
</REQUEST>
</ARCXML>
```

Example 4: When in SPATIALFILTER in CONFIG or REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_IMAGE>
    <PROPERTIES>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      <IMAGESIZE width="643" height="502" />
    </PROPERTIES>
    <LAYER type="featureclass" name="select layer" visible="true"
id="selected">
      <DATASET fromlayer="countries" />
      <SPATIALQUERY>
        <SPATIALFILTER relation="area_intersection">
          <POLYGON>
            <RING>
              <POINT x="83.15605550814075" y="38.07185101549165" />
              <POINT x="111.09942196116728" y="-4.70645066589869" />
              <POINT x="155.1079549837513" y="-10.38915084069517" />
              <POINT x="139.1079549837513" y="66.38915084069517" />
              <POINT x="83.15605550814075" y="38.07185101549165" />
            </RING>
          </POLYGON>
        </SPATIALFILTER>
      </SPATIALQUERY>
      <SIMPLERENDERER>
        <SIMPLEPOLYGONSYMBOL fillcolor="0,0,0" filltype="cross" />
      </SIMPLERENDERER>
    </LAYER>
  </GET_IMAGE>
</REQUEST>
</ARCXML>
```

Example 5: When in MARKUP.

```
<?xml version="1.0"?>
<MARKUP>
  <WORKSPACES>
    <FEATURESERVERWORKSPACE name="ifs_ws-0"
url="http://mycomputer.domain.com/servlet/com.esri.esrimap.Esrimap"
service="world" />
  </WORKSPACES>
  <MARKUPLAYER layername="Countries" workspace="ifs_ws-0">
    <ADDEDFEATURES>
      <FEATURE featureid="1000000">
      <ENVELOPE minx="-35.92572247220596" miny="-48.809104594884616"
```

```
maxx="-5.291989044369359" maxy="-21.047283675907693" />
      <FIELD name="AREA" precision="3" size="12" type="8" />
      <FIELD name="NAME" precision="0" size="40" type="12" />
      <FIELD name="ABBREVNAME" precision="0" size="12" type="12" />
      <FIELD name="FIPS_CODE" precision="0" size="2" type="12" />
      <FIELD name="WB_CNTRY" precision="0" size="3" type="12" />
      <FIELD name="#SHAPE#" precision="0" size="0" type="-98">
        <FIELDVALUE>
          <POLYGON>
            <RING>
            <POINT x="-19.17289950385782" y="-21.047283675907693" />
            <POINT x="-35.92572247220596" y="-38.2787587290658" />
            <POINT x="-10.078509892468816" y="-48.809104594884616" />
            <POINT x="-5.291989044369359" y="-24.397848269577324" />
            <POINT x="-19.17289950385782" y="-21.047283675907693" />
            </RING>
          </POLYGON>
        </FIELDVALUE>
      </FIELD>
      </FEATURE>
    </ADDEDFEATURES>
  </MARKUPLAYER>
</MARKUP>
```

POLYLINE

Used in: **CONFIG** **REQUEST** **RESPONSE** **MARKUP**
Servers: Image Query Feature Extract ArcMap
Parent elements: **FEATURE** **FIELDVALUE** **OBJECT** **SPATIALFILTER**

<POLYLINE >
No Attributes

(m) <PATH... />

</POLYLINE >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Provides the x,y coordinate locations for a line feature.

Restrictions:

None

Notes:

- For the different options of representing geometry in an acetate layer, see OBJECT.

Examples for POLYLINE:

Example 1: When in SPATIALFILTER in CONFIG or REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <IMAGESIZE width="643" height="502" />
      </PROPERTIES>
      <LAYER type="featureclass" name="select layer" visible="true"
id="Selected">
        <DATASET fromlayer="1" />
        <SPATIALQUERY>
          <SPATIALFILTER relation="area_intersection">
            <POLYLINE>
              <PATH>
                <POINT x="-2.1079549837513" y="19.99815142335011" />
                <POINT x="28.99468788980437" y="15.88488432940315" />
                <POINT x="55.99468788980437" y="35.88488432940315" />
              </PATH>
            </POLYLINE>
          </SPATIALFILTER>
```

```
        </SPATIALQUERY>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="0,255,0" filltype="cross"
fillinterval="3" />
        </SIMPLERENDERER>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 2: When using POLYLINE in an acetate layer.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <IMAGESIZE width="800" height="600" />
      </PROPERTIES>
      <LAYER type="acetate" name="Selectedmark" id="acetate">
        <OBJECT units="database">
          <SIMPLELINESYMBOL color="0,0,0" width="6" />
          <POLYLINE>
            <PATH>
              <POINT x="-2.1079549837513" y="19.99815142335011" />
              <POINT x="28.99468788980437" y="15.88488432940315" />
              <POINT x="55.99468788980437" y="35.88488432940315" />
            </PATH>
          </POLYLINE>
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 3: When in a FEATURES response.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
<RESPONSE>
  <FEATURES>
    <FEATURE>
      <FIELDS CUST_ID="4" NAME="Customer  4"  #SHAPE#="[Geometry]"
#ID#="3" />
      <POLYLINE>
        <PATH>
          <POINT x="-128.1079549837513" y="81.99815142335011" />
          <POINT x="-123.99468788980437" y="77.88488432940315" />
          <POINT x="-123.99468788980437" y="77.88488432940315" />
        </PATH>
      </POLYLINE>
    </FEATURE>
    <FEATURECOUNT count="1" hasmore="false" />
```

```
  </FEATURES>
</RESPONSE>
</ARCXML>
```

PROPERTIES

Used in: CONFIG REQUEST RESPONSE
Servers: Image Extract ArcMap
Parent elements: DATAFRAMEINFO GET_EXTRACT GET_IMAGE GET_LAYOUT MAP SERVICEINFO

<PROPERTIES >
No Attributes

*When parent element is **DATAFRAMEINFO:***
<ENVELOPE... />
<FEATURECOORDSYS... />
<FILTERCOORDSYS... />
<MAPUNITS... />

*When parent element is **GET_EXTRACT:***
<ENVELOPE... />
<FEATURECOORDSYS... />
<FILTERCOORDSYS... />
<IMAGESIZE... />
<LAYERLIST... />
<OUTPUT... />

*When parent element is **GET_IMAGE:***
<BACKGROUND... />
<DRAW... />
<ENVELOPE... />
<FEATURECOORDSYS... />
<FILTERCOORDSYS... />
<IMAGESIZE... />
<LAYERLIST... />
<LEGEND... />
<OUTPUT... />

*When parent element is **GET_LAYOUT:***
<ENVELOPE... />
<FEATURECOORDSYS... />
<FILTERCOORDSYS... />
<IMAGESIZE... />
<OUTPUT... />

*When parent element is **MAP:***
<ENVELOPE... />
<FEATURECOORDSYS... />
<FILTERCOORDSYS... />

<IMAGEGENERALIZATION... />
<MAPUNITS... />
<OUTPUT... /> *[Image Server and Extract Server only]*
<BACKGROUND... /> *[Image Server only]*
<LEGEND... /> *[Image Server only]*

*When parent element is **SERVICEINFO:***
<ENVELOPE... />
<FEATURECOORDSYS... />
<FILTERCOORDSYS... />
<IMAGEGENERALIZATION... />
<MAPUNITS... />
<BACKGROUND... /> *[Image Server and ArcMap Server only]*
<IMAGESIZE... /> *[Image Server only]*
<LEGEND... /> *[Image Server only]*
<OUTPUT... /> *[Image Server only]*

</PROPERTIES >

Bold: Attribute or child element is required.

Description:

Provides the framework for defining properties about an ArcIMS service. In requests, it provides the framework for modifying properties information in a map configuration file.

Restrictions:

- When using MAP, the child element OUTPUT is valid only with the Image and Extract Servers.
- When using MAP, the child elements BACKGROUND and LEGEND are valid only with the Image Server.
- When using SERVICEINFO, the child elements IMAGESIZE, LEGEND, and OUTPUT are valid only with the Image Server.
- When using SERVICEINFO, the child element BACKGROUND is valid only with the Image Server.

Notes:

None

Examples for PROPERTIES:

Example 1: When in CONFIG.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.21" miny="18.92" maxx="-66.96"
maxy="71.41" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CITIES" visible="true" id="2">
      <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLEMARKERSYMBOL type="square" width="5" />
      </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When in a GET_IMAGE or GET_EXTRACT request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
  <GET_IMAGE>
    <PROPERTIES>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      <LAYERLIST>
        <LAYERDEF id="Cities">
          <SIMPLERENDERER>
            <SIMPLEMARKERSYMBOL size="16" color="0,0,0" />
          </SIMPLERENDERER>
          <SPATIALQUERY>
            <SPATIALFILTER relation="area_intersection">
              <ENVELOPE maxy="60" maxx="60" miny="0" minx="0" />
            </SPATIALFILTER>
          </SPATIALQUERY>
        </LAYERDEF>
      </LAYERLIST>
    </PROPERTIES>
  </GET_IMAGE>
  </REQUEST>
```

```
</ARCXML>
```

Example 3: When in a SERVICEINFO response.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-71.0718204242754" miny="42.368904975182"
maxx="-71.0475995680561" maxy="42.3869647980717" name="Initial_Extent"
/>
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="Streets"
id="1" >
        <FCLASS type="line"></FCLASS>
        <EXTENSION type="Geocode" >
          <GCSTYLE name="USAddressZ" />
        </EXTENSION>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

PUBLISH_METADATA

Used in: **REQUEST**
Servers: Metadata (Publish)
Parent elements: **REQUEST**

<PUBLISH_METADATA >
No Attributes

<CHANGE_METADATA_ACCESS... /> *[Or]*
<DELETE_METADATA... /> *[Or]*
<DELETE_METADATA_RELATIONSHIP... /> *[Or]*
<GET_UUID... /> *[Or]*
<PUT_METADATA... /> *[Or]*
<PUT_METADATA_RELATIONSHIP... /> *[Or]*
<PUT_METADATA_SEMANTIC... /> *[Or]*
<PUT_USER... /> *[Or]*
<RENAME_METADATA... /> *[Or]*
<RESET... /> *[Or]*

</PUBLISH_METADATA >

Bold: Attribute or child element is required.

Description:

Request for administering and publishing metadata.

Restrictions:

- Only one child element can be used per request.

Notes:

None

Examples for PUBLISH_METADATA:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <PUBLISH_METADATA>
      <PUT_METADATA name="World" userid="{C64D8F38-82B4-11D5-99C2-
000086460FA0}" private="false" folder="true" parentname="Home"
parentowner="admin1" parentuserid="root" >
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <THUMBNAIL>...</THUMBNAIL>
        <!--<?xml version="1.0"?>
            <metadata>...</metadata>-->
      </PUT_METADATA>
```

```
    </PUBLISH_METADATA>
  </REQUEST>
</ARCXML>
```

PUT_METADATA

Used in: REQUEST
Servers: Metadata (Publish)
Parent elements: PUBLISH_METADATA

```
<PUT_METADATA

    docid ="string"
    name ="string"
    content ="liveData | downloadableData | offlineData | staticMapImage | document |
application | geographicService | clearinghouse"
    folder ="true | false"  [false]
    onlink ="string"
    parentdocid ="string"
    private ="true | false"  [false]
    server ="string"
    service ="string"
    servicetype ="metadata | image | feature | wms"
>

    <ENVELOPE... />
    <THUMBNAIL... />

</PUT_METADATA >
```

Bold: Attribute or child element is required.

Description:

Publishes a metadata document to the Metadata Server.

Restrictions:

None

Notes:

- See METADATA_ACTION for response.
- The metadata document is sent inside PUT_METADATA enclosed in an XML comment:

  <!-- content -->

  The enclosed document must have a processing instruction, must not specify encoding, and cannot contain any comments since comments within comments are illegal.

  <PUT_METADATA>

```
<!-- metadata document -->
</PUT_METADATA>
```

- If a document with the same name and document ID is republished, then the old document is deleted and the new document is inserted. This results in the internal ID changing. In this case, if an administration table exists, an update query is issued against the table to update the internal ID in the administration table's join field.
- The document owner is implicitly defined by the logged in user who is adding the document.

Attribute Descriptions for PUT_METADATA:

<table>
<tr><th>Attribute</th><th>Usage</th></tr>
<tr><td>content</td><td>The values for this attribute correspond to the content types listed in the drop-down comb box of the Metadata Explorer.
<table>
<tr><th>ArcXML Attribute Value</th><th>Combo Box Content</th></tr>
<tr><td>liveData</td><td>Live Data and Maps</td></tr>
<tr><td>downloadableData</td><td>Downloadable Data</td></tr>
<tr><td>offlineData</td><td>Offline Data</td></tr>
<tr><td>staticMapImage</td><td>Static Map Images</td></tr>
<tr><td>documents</td><td>Other Documents</td></tr>
<tr><td>application</td><td>Applications</td></tr>
<tr><td>geographicService</td><td>Geographic Services</td></tr>
<tr><td>clearinghouse</td><td>Clearinghouses</td></tr>
</table></td></tr>
<tr><td>docid</td><td>String used to uniquely identify a document. The client used to publish the metadata is responsible for creating the document ID. This ID is automatically assigned when using ArcCatalog. If another client is used, GET_UUID can be used to request a valid ID. The format for an ID is the following: {HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH} where H is a hexadecimal digit (0-9,a-f,A-F). The ID is limited to 38 characters.</td></tr>
<tr><td>folder</td><td>Determines whether document represents a folder. Use "false" if the document does not contain other documents. Use "true" if the document represents a folder.</td></tr>
<tr><td>name</td><td>Name that identifies the dataset corresponding to the document.</td></tr>
<tr><td>onlink</td><td>A string identifying the location of the dataset corresponding to the document.</td></tr>
</table>

parentdocid	Unique string for identifying a document belonging to the parent name and parent owner.
private	Determines whether document is viewable for all users. Use "false" if the document is viewable by all users. Use "true" when the document is viewable only by the document owner.
server	The URL of the Web server containing data or metadata associated with the document, for example, http://mymachine.domain.com.
service	The name of the service containing data or metadata associated with the document.
servicetype	Type of ArcIMS service.

Examples for PUT_METADATA:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <PUBLISH_METADATA>
      <PUT_METADATA docid="{C64D8F38-82B4-11D5-99C2-000086460FA0}"
private="false" folder="true" parentdocid="root" >
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <THUMBNAIL>...</THUMBNAIL>
        <!--<?xml version="1.0"?>
            <metadata>...</metadata>-->
      </PUT_METADATA>
    </PUBLISH_METADATA>
  </REQUEST>
</ARCXML>
```

PUT_METADATA_RELATIONSHIP

Used in: REQUEST
Servers: Metadata (Publish)
Parent elements: PUBLISH_METADATA

<PUT_METADATA_RELATIONSHIP >
No Attributes

<METADATA_SOURCE... />
(m) <METADATA_CHILD... /> *[And/Or]*
(m) <METADATA_SIBLING... /> *[And/Or]*

</PUT_METADATA_RELATIONSHIP >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Establishes a relationship between a source metadata document and one or more child or sibling metadata documents.

Restrictions:

- At least one occurrance of METADATA_CHILD or METADATA_SIBLING is required. One or both elements can be used multiple times.

Notes:

- See METADATA_ACTION for response.
- A child document is a subdocument to the current metadata document. A sibling document is a related document. When selected as a sibling, the document is listed with the parent document under a link called "related records".

Examples for PUT_METADATA_RELATIONSHIP:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <PUBLISH_METADATA>
      <PUT_METADATA_RELATIONSHIP>
        <METADATA_SOURCE docid="{C64D8F38-82B4-11D5-99C2-000086460FA0}"
/>
        <METADATA_CHILD docid="{F99A8C01-76B3-62E5-12G4-056786460CA0}"
/>
      </PUT_METADATA_RELATIONSHIP>
    </PUBLISH_METADATA>
  </REQUEST>
</ARCXML>
```

PUT_METADATA_SEMANTIC

Used in: REQUEST
Servers: Metadata (Publish)
Parent elements: PUBLISH_METADATA

<PUT_METADATA_SEMANTIC >
No Attributes

(m) <SEMANTIC_PAIR... />

</PUT_METADATA_SEMANTIC >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Framework for establishing a relationship between a tag and the numeric code defining semantics from the GEO Attribute Set.

Restrictions:

None

Notes:

- See METADATA_ACTION for response.

Examples for PUT_METADATA_SEMANTIC:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <PUBLISH_METADATA>
      <PUT_METADATA_SEMANTIC>
        <SEMANTIC_PAIR tag="idinfo/citation/citeinfo/title" zcode="4"
/>
      </PUT_METADATA_SEMANTIC>
    </PUBLISH_METADATA>
  </REQUEST>
</ARCXML>
```

PUT_USER

Used in: REQUEST
Servers: Metadata (Publish)
Parent elements: PUBLISH_METADATA

<PUT_USER >
No Attributes

No Child Elements
</PUT_USER >

Description:

Associates an XML document with the current user.

Restrictions:

- Not available from ArcCatalog.

Notes:

- See METADATA_ACTION for response.
- The user information is sent inside PUT_USER enclosed in an XML comment:

  <!-- content -->

  The enclosed document must have a processing instruction, must not specify encoding, and cannot contain any comments since comments within comments are illegal.

  <PUT_USER>
  <!-- user info -->
  </PUT_USER>

Examples for PUT_USER:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <PUBLISH_METADATA>
      <PUT_USER>
        <!--<?xml version="1.1"?><USER name="aUser"
url="http://www.esri.com/output" />-->
      </PUT_USER>
    </PUBLISH_METADATA>
  </REQUEST>
</ARCXML>
```

QUERY

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Query Feature Extract ArcMap
Parent elements: **GET_FEATURES** **LAYER** **LAYERDEF** **STOREDQUERY**

<QUERY

When using ArcMap Server:
where ***="string"***
featurelimit ="*integer*" **[All that meet criteria]**
subfields ="#ALL# | #ID# | #SHAPE# | Other fields in database" **[#ALL#]**

When using Image, Extract, Query, or Feature Server:
where ***="string"***
accuracy ="Distance between points" **[0]**
featurelimit ="*integer*" **[All that meet criteria]**
joinexpression ="*string*"
jointables ="*string*"
subfields ="#ALL# | #ID# | #SHAPE# | Other fields in database" **[#ALL#]**
>

When parent element is ***GET_FEATURES:***
<BUFFER... />
<FEATURECOORDSYS... />

When parent element is ***LAYER and LAYERDEF in GET_IMAGE:***
<BUFFER... />

</QUERY>

Bold: Attribute or child element is required.

Description:

- Defines the *where* clause for a stored query.
- Defines attribute queries on a dataset.

Restrictions:

- A *where* statement is required for all stored queries and when joining ArcSDE tables. Otherwise, *where* is optional.

Notes:

- For information on joining tables and setting up query clauses, see SPATIALQUERY. QUERY handles only attributes, while SPATIALQUERY handles both attribute and spatial queries. It is recommended that you use SPATIALQUERY for all query statements with the exception of stored queries.

- QUERY must be used when setting up a stored query.
- Joined tables cannot be used with a QUERY in STOREDQUERY.
- When ArcSDE is used, the *subfields* attribute has the full long format of the field names (e.g., MD.GDT_STATE.AREA). For shapefiles, the *subfields* attribute contains the field names in short format (e.g., AREA).
- In a stored query, only one variable can be used.
- Some symbols must be "escaped" inside a *where* expression:

  ampersand (&) is escaped to &
  double quotes (") are escaped to "
  single quotes (') are escaped to '
  greater than (>) is escaped to >
  less than (<) is escaped to <
- The following operators work in a where clause: =, >, >=, <, <=, <>, LIKE, BETWEEN, IN, NOT IN. The following are not valid: ORDER BY and DISTINCT.
- **Querying with dates**. The syntax for querying dates is the same regardless of the locale. A date query uses the following syntax:

  {ts 'YYYY-MM-DD hh:mi:ss'}

  where

YYYY	Year	Required	Use four digits for the year.
MM	Month (01-12)	Required	Use two digits for the month. March is 03.
DD	Day (01-31)	Required	Use two digits for the day. The fourth is 04.
hh	Hour (00-23)	Optional	Use a 24-hour clock. 8 a.m. is 08, and 8 p.m. is 20.
mi	Minutes (00-59)	Optional	Use two digits for the minutes. If minutes is used, hours must also be included.
ss	Seconds (00-59)	Optional	Use two digits for the seconds. If seconds is used, hours and minutes must also be included.

  The year, month, and day are each separated by a dash (-). The hour, minutes, and seconds are each separated by a colon (:). The date is enclosed in single quotes (') inside curly brackets ({}). Before the date, ts (for time stamp) must be included.

  For 8:03:23 a.m. January 4, 2000, the query on a DBF file looks like:
  <QUERY where="MYDATE = {ts '2000-01-04 08:03:32'}" />

  For 9:18 p.m. March 8, 2002, the query on an ArcSDE layer looks like:
  <QUERY where="ARCSDE.TABLE.MYDATE = {ts '2002-03-08 21:18:00'}" />

Attribute Descriptions for QUERY:

Attribute	Usage
accuracy	Within a feature, generalizes points based on the distance specified and the resolution of the image.
featurelimit	Maximum number of features to be extracted that meet criteria.
joinexpression	Used for join tables with DBF files only; *jointables* must be filled to contain list of tables used; not required when a jointable is done on ArcSDE. String must form expression: "To=[master table column which will be used for joining], From=[defines a join table column which will be joined], Type=[exact or scan]".
jointables	List of joined table names separated by blank spaces; for ArcSDE, table name is full name including database name (e.g., DATA.STATE); for shapefiles, names of DBF files without extension (e.g., STATES).
subfields	List of fields to be extracted separated by blank space. In stored queries, all fields in the layer table must be included.
where	Defines where part of SQL expression. Required when *jointables* attribute for ArcSDE tables is used.

Examples for QUERY:

Example 1: When in a map configuration file using a STOREDQUERY. For additional examples on querying datasets in map configuration files and requests, see SPATIALQUERY.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.21" miny="18.92" maxx="-66.96"
maxy="71.41" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CITIES" visible="true" id="2">
      <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
      <SIMPLERENDERER>
```

```
        <SIMPLEMARKERSYMBOL type="square" width="5" />
      </SIMPLERENDERER>
        <EXTENSION type="StoredQuery">
        <STOREDQUERIES>
          <STOREDQUERY name="TestSt">
            <QUERY where=" ZIPL  =  &apos;[%var%]&apos; "
subfields="#SHAPE# FNODE_ TNODE_ LPOLY_ RPOLY_ LENGTH RECNUM L_F_ADD
L_T_ADD R_F_ADD R_T_ADD PREFIX NAME TYPE SUFFIX CFCC ZIPL ZIPR"/>
            <SQVAR position="0" name="[%var%]">
              <FIELD name="ZIPL" precision="0" type="12" size="5" />
            </SQVAR>
          </STOREDQUERY>
        </STOREDQUERIES>
        </EXTENSION>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

RANGE

Used in: CONFIG REQUEST RESPONSE
Servers: Image Feature
Parent elements: VALUEMAPLABELRENDERER VALUEMAPRENDERER

<RANGE

lower ="***string, numeric, or date***"
upper ="***string, numeric, or date***"
equality ="all | upper | lower | none" **[lower]**
label ="*string*"
>

*When parent element is **VALUEMAPLABELRENDERER:***
<CALLOUTMARKERSYMBOL... /> *[Or]*
<CHARTSYMBOL... /> *[Or]*
<RASTERSHIELDSYMBOL... /> *[Or]*
<SHIELDSYMBOL... /> *[Or]*
<TEXTSYMBOL... /> *[Or]*

*When parent element is **VALUEMAPRENDERER:***
<GRADIENTFILLSYMBOL... /> *[Or]*
<HASHLINESYMBOL... /> *[Or]*
<RASTERFILLSYMBOL... /> *[Or]*
<RASTERMARKERSYMBOL... /> *[Or]*
<SIMPLELINESYMBOL... /> *[Or]*
<SIMPLEMARKERSYMBOL... /> *[Or]*
<SIMPLEPOLYGONSYMBOL... /> *[Or]*
<TRUETYPEMARKERSYMBOL... /> *[Or]*

</RANGE >

Bold: Attribute or child element is required.

Description:

Used with VALUEMAPRENDERER and VALUEMAPLABELRENDERER for matching a range of values within a specified field in a database. When a match occurs, the symbol is drawn as specified for each range.

Restrictions:

- One renderer child element is required for each RANGE element in the value map.
- Not valid with ArcMap Server.

Notes:

- If DATASET layer type is point, then only point symbols can be used. If type is line, then line and point symbols can be used. If type is polygon, then polygon, line, and point symbols can be used.
- If there are leading or trailing blanks in a field value, they are trimmed before a comparison is made. For example, a field value of " Hello " is interpreted as "Hello".
- For more information on using renderers, see Using ArcXML Renderers.

Attribute Descriptions for RANGE:

Attribute	Usage
equality	Defines the upper and lower bounds of each range. If "all" is used, then lower <= value <= upper. If "upper" is used, lower < value <= upper. If "lower" is used, lower <= value < upper.
label	Label for legend.
lower	Lower value of range; can be a numeric, string, or date value.
upper	Upper value of range; can be a numeric, string, or date value.

Examples for RANGE:

Example 1: When using VALUEMAPRENDERER.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="Countries" visible="true"
id="100">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <VALUEMAPRENDERER lookupfield="AREA">
        <RANGE lower="0.0" upper="1000000.0" equality="all"
label="Small">
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,255,0"
boundarywidth="1" />
        </RANGE>
        <RANGE lower="1000000.0" upper="3000000.0" equality="upper"
```

```
label="Medium">
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,146,0"
boundarywidth="2" />
        </RANGE>
        <RANGE lower="3000000.0" upper="10000000.0" equality="upper"
label="Large">
          <RASTERMARKERSYMBOL shadow="0,0,0" overlap="true"
url="http://arcims2/website/color.gif"
image="C:\ArcIMS\WebSite\color.gif" />
        </RANGE>
        <OTHER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="128,128,128"
/>
        </OTHER>
        </VALUEMAPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When using VALUEMAPLABELRENDERER.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <<LAYER type="featureclass" name="Cities" visible="true" id="1">
        <DATASET name="cities" type="point" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <VALUEMAPLABELRENDERER lookupfield="POPULATION"
labelfield="NAME" >
          <RANGE lower="0.0" upper="150000.0">
            <TEXTSYMBOL font="Tahoma" fontstyle="regular" fontsize="10"
/>
          </RANGE>
          <RANGE lower="150001.0" upper="750000.0">
            <TEXTSYMBOL font="Arial" fontstyle="italic" fontsize="12"
glowing="125,125,125" />
          </RANGE>
          <RANGE lower="750001.0" upper="3427180.0">
            <TEXTSYMBOL font="Times New Roman" fontstyle="bolditalic"
```

```
fontsize="14" glowing="255,255,0" shadow="0,0,0" />
          </RANGE>
          </VALUEMAPLABELRENDERER>
          <SIMPLERENDERER>
            <SIMPLEMARKERSYMBOL color="51,102,51" width="8" />
          </SIMPLERENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

RASTER_EXACT

Used in: CONFIG
Parent elements: RASTER_RENDERER

<RASTER_EXACT

value ="***double***"
color ="0,0,0 - 255,255,255" **[0,0,0]**
label ="*string*" **[Value of index]**
transparency ="0.0 - 1.0" **[1.0]**
>

No Child Elements
</RASTER_EXACT >

Bold: Attribute or child element is required.

Description:

Defines description and color for one pixel value.

Restrictions:

- RASTER_EXACT is not available from the ArcIMS Author interface.
- Raster renderering is available in all viewers. However, labels in the legend are valid only in HTML Viewers.
- A value cannot be duplicated in a raster renderer. If a duplicate is detected, an error message is returned.
- Valid with Image Server only.

Notes:

- Three methods are available to define colors in an image.
  - ArcSDE colormap or CLR file, which pairs a value and a color. The value is included in the legend.
  - Table in ArcSDE that pairs a value with a color and description. The description is included in the legend. If no description is available, the value is used instead. If a table is used, the colormap is ignored.
  - The RASTER_RENDERER elements. If these elements are used, any colormaps or ArcSDE tables are ignored.

Attribute Descriptions for RASTER_EXACT:

Attribute	Usage
color	Color using RGB values. Defines the color corresponding to the specified index value.
label	Description to be shown on legend. Valid with HTML Viewers only.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.
value	Index value in the raster image. Can be image or floating point values.

Examples for RASTER_EXACT:

Example 1: When in a map configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="0" miny="0" maxx="256" maxy="256"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SDEWORKSPACE name="sde_ws-0"  server="myserver"
instance="port:5151" database="" user="WASHOE" encrypted="true"
password="LXEMUR" />
      </WORKSPACES>
      <LAYER type="image" name="Land Use" visible="true" id="0">
        <DATASET name="WASHOE.BIT8_COLORMAP.IMAGE" workspace="sde_ws-0"
/>
        <RASTER_RENDERER>
          <RASTER_EXACT value="0" color="10,200,10"
transparency="0.5"  label="Clouds"/>
          <RASTER_RANGE lower="1" upper="101" color="200,40,10"
transparency="0.5"  equality="lower" label="Urban"/>
          <RASTER_RANGE transparency="1.0" lower="101" upper="255"
color="255,255,10" equality="lower" label="Rural"/>
          <RASTER_OTHER transparency="0.0" label="Other"/>
        </RASTER_RENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

RASTER_INFO

Used in: **RESPONSE**
Servers: Image ArcMap
Parent elements: **RESPONSE**

<RASTER_INFO >
No Attributes

(m) <BANDS... />

</RASTER_INFO >

(m): Child element can be used multiple times.

Description:

Parent element for returning pixel values of a raster image at a specified x,y coordinate.

Restrictions:

None

Notes:

None

Examples for RASTER_INFO:

Example 1:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <RASTER_INFO>
      <BANDS rasterid="1">
        <BAND number="0" value="238" />
      </BANDS>
    </RASTER_INFO>
  </RESPONSE>
</ARCXML>
```

RASTER_OTHER

Used in: CONFIG
Parent elements: RASTER_RENDERER

```
<RASTER_OTHER

    color ="0,0,0 - 255,255,255"  [0,0,0]
    label ="string"  ["Out of Range"]
    transparency ="0.0 - 1.0"  [1.0]
>

    No Child Elements
</RASTER_OTHER >
```

Description:

Defines description and color for all pixel values not covered by RASTER_EXACT and RASTER_RANGE.

Restrictions:

- RASTER_OTHER is not available from the ArcIMS Author interface.
- Raster renderering is available in all viewers. However, labels in the legend are valid only in HTML Viewers.
- A value cannot be duplicated in a raster renderer. If a duplicate is detected, an error message is returned.
- Valid with Image Server only.

Notes:

- Three methods are available to define colors in an image.
    - ArcSDE colormap or CLR file, which pairs a value and a color. The value is included in the legend.
    - Table in ArcSDE that pairs a value with a color and description. The description is included in the legend. If no description is available, the value is used instead. If a table is used, the colormap is ignored.
    - The RASTER_RENDERER elements. If these elements are used, any colormaps or ArcSDE tables are ignored.

Attribute Descriptions for RASTER_OTHER:

Attribute	Usage
color	Color using RGB values. Defines the color for all index values not previously defined.
label	Description to be shown on legend. Valid with HTML Viewers only.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.

Examples for RASTER_OTHER:

Example 1: When in a map configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="0" miny="0" maxx="256" maxy="256"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SDEWORKSPACE name="sde_ws-0"  server="myserver"
instance="port:5151" database="" user="WASHOE" encrypted="true"
password="LXEMUR" />
      </WORKSPACES>
      <LAYER type="image" name="Land Use" visible="true" id="0">
        <DATASET name="WASHOE.BIT8_COLORMAP.IMAGE" workspace="sde_ws-0"
/>
        <RASTER_RENDERER>
          <RASTER_EXACT value="0" color="10,200,10"
transparency="0.5"  label="Clouds"/>
          <RASTER_RANGE lower="1" upper="101" color="200,40,10"
transparency="0.5"  equality="lower" label="Urban"/>
          <RASTER_RANGE transparency="1.0" lower="101" upper="255"
color="255,255,10" equality="lower" label="Rural"/>
          <RASTER_OTHER transparency="0.0" label="Other"/>
        </RASTER_RENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

RASTER_RANGE

Used in: CONFIG
Parent elements: RASTER_RENDERER

<RASTER_RANGE

lower ="***double***"
upper ="***double***"
color ="0,0,0 - 255,255,255" **[0,0,0]**
equality ="all | upper | lower | none" **[lower]**
label ="*string*" **[Value of lower or upper]**
transparency ="0.0 - 1.0" **[1.0]**
>

No Child Elements
</RASTER_RANGE >

Bold: Attribute or child element is required.

Description:

Defines description and color for a range of pixel values.

Restrictions:

- RASTER_RANGE is not available from the ArcIMS Author interface.
- Raster renderering is available in all viewers. However, labels in the legend are valid only in HTML Viewers.
- A value cannot be duplicated in a raster renderer. If a duplicate is detected, an error message is returned.
- Valid with Image Server only.

Notes:

- Three methods are available to define colors in an image.
  - ArcSDE colormap or CLR file, which pairs a value and a color. The value is included in the legend.
  - Table in ArcSDE that pairs a value with a color and description. The description is included in the legend. If no description is available, the value is used instead. If a table is used, the colormap is ignored.
  - The RASTER_RENDERER elements. If these elements are used, any colormaps or ArcSDE tables are ignored.

Attribute Descriptions for RASTER_RANGE:

Attribute	Usage
color	Color using RGB values. Defines the color corresponding to the specified range of index values.
equality	Defines the upper and lower bounds of each range. If "all" is used, then

	lower <= value <= upper. If "upper" is used, lower < value <= upper. If "lower" is used, lower <= value < upper.
label	Description to be shown on legend. Valid with HTML Viewers only.
lower	Defines the lower value of the range.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.
upper	Defines the upper value of the range.

Examples for RASTER_RANGE:

Example 1: When in a map configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="0" miny="0" maxx="256" maxy="256"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SDEWORKSPACE name="sde_ws-0"  server="myserver"
instance="port:5151" database="" user="WASHOE" encrypted="true"
password="LXEMUR" />
      </WORKSPACES>
      <LAYER type="image" name="Land Use" visible="true" id="0">
        <DATASET name="WASHOE.BIT8_COLORMAP.IMAGE" workspace="sde_ws-0"
/>
        <RASTER_RENDERER>
          <RASTER_EXACT value="0" color="10,200,10"
transparency="0.5"  label="Clouds"/>
          <RASTER_RANGE lower="1" upper="101" color="200,40,10"
transparency="0.5"  equality="lower" label="Urban"/>
          <RASTER_RANGE transparency="1.0" lower="101" upper="255"
color="255,255,10" equality="lower" label="Rural"/>
          <RASTER_OTHER transparency="0.0" label="Other"/>
        </RASTER_RENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

RASTER_RENDERER

Used in: CONFIG
Parent elements: LAYER

<RASTER_RENDERER >
No Attributes

(m) <RASTER_EXACT... />
<RASTER_OTHER... />
(m) <RASTER_RANGE... />

</RASTER_RENDERER >

(m): Child element can be used multiple times.

Description:

Renders images by classifying pixel values.

Restrictions:

- RASTER_RENDERER is not available from the ArcIMS Author interface.
- Raster renderering is available in all viewers. However, labels in the legend are valid only in HTML Viewers.
- A value cannot be duplicated in a raster renderer. If a duplicate is detected, an error message is returned.
- Valid with Image Server only.

Notes:

- Three methods are available to define colors in an image.
    - ArcSDE colormap or CLR file, which pairs a value and a color. The value is included in the legend.
    - Table in ArcSDE that pairs a value with a color and description. The description is included in the legend. If no description is available, the value is used instead. If a table is used, the colormap is ignored.
    - The RASTER_RENDERER elements. If these elements are used, any colormaps or ArcSDE tables are ignored.

Examples for RASTER_RENDERER:

Example 1: When in a map configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="0" miny="0" maxx="256" maxy="256"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SDEWORKSPACE name="sde_ws-0"  server="myserver"
instance="port:5151" database="" user="WASHOE" encrypted="true"
password="LXEMUR" />
      </WORKSPACES>
      <LAYER type="image" name="Land Use" visible="true" id="0">
        <DATASET name="WASHOE.BIT8_COLORMAP.IMAGE" workspace="sde_ws-0"
/>
        <RASTER_RENDERER>
          <RASTER_EXACT value="0" color="10,200,10"
transparency="0.5"  label="Clouds"/>
          <RASTER_RANGE lower="1" upper="101" color="200,40,10"
transparency="0.5"  equality="lower" label="Urban"/>
          <RASTER_RANGE transparency="1.0" lower="101" upper="255"
color="255,255,10" equality="lower" label="Rural"/>
          <RASTER_OTHER transparency="0.0" label="Other"/>
        </RASTER_RENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

RASTERFILLSYMBOL

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Feature ArcMap
Parent elements: **EXACT** **OBJECT** **OTHER** **POLYGON** **RANGE** **SIMPLERENDERER**

<RASTERFILLSYMBOL

When using ArcMap Server:
url ="url string"
image ="path to image file"

When using Image or Feature Server:
url ="url string"
antialiasing ="true | false" **[false]**
image ="path to image file"
overlap ="true | false" **[true]**
transparency ="0.0 - 1.0" **[1.0]**
>

No Child Elements
</RASTERFILLSYMBOL >

Bold: Attribute or child element is required.

Description:

Fills polygon features with specified image.

Restrictions:

- *Overlap* is valid only for Image Services. It is ignored when using Feature Services.
- When using an ArcIMS HTML Viewer, ArcIMS Java Viewer, ArcExplorer 4, or any other browser using the ArcIMS Servlet Connector, RASTERFILLSYMBOL is restricted in a SERVICEINFO response. The attribute *image* is not returned. This restriction can be lifted by setting the property *spatialServer.AllowResponsePath* to "true" in esrimap_prop. This property file is found in the servlet engine directory. For more information on the location of esrimap_prop and its properties, see the ArcIMS Help.

  This restriction applies only when the ArcIMS Servlet Connector is used. It does not apply to the ActiveX, ColdFusion, or Java Connectors.

- In ArcMap Image Services, symbol is valid only in acetate layers.

Notes:

- Acceptable image formats are JPG and GIF.
- In the legend of ArcIMS Java Viewers and ArcExplorer 4, the swatch showing a symbol is limited in size. Images greater than approximately 16x16 pixels in size are truncated to fill the swatch. The full image displays on the map. In the ArcIMS HTML Viewer and other HTML implementations, the size of the swatch can be controlled using LEGEND.

Attribute Descriptions for RASTERFILLSYMBOL:

Attribute	Usage
antialiasing	Used to make edges of labels and symbols smoother. When set to "true", antialiasing is active. Note that the time to generate a map may significantly increase when antialiasing is turned on.
image	Full pathname to image. ArcIMS Spatial Server uses this pathname to find the image and add it to the map. UNC pathnames can be used (\\myComputer\arcims\output).
overlap	Determines if labels can overlap this symbol. When "true", labels can overlap. When "false", labels will not overlap the symbol. If labels are not drawing as expected, check if *overlap* is set to "false" for this symbol or any other symbol in the ArcIMS service.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.
url	URL used by client to retrieve image.

Examples for RASTERFILLSYMBOL:

Example 1: When in CONFIG or REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-141.003006" miny="41.913319" maxx="-52.620281"
maxy="83.108322" name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to CANADA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="province" visible="true" id="0">
        <DATASET name="province" type="polygon" workspace="shp_ws-0" />
```

```
        <SIMPLERENDERER>
          <RASTERFILLSYMBOL transparency="0.5" overlap="true"
url="http://mymachine.domain.com/website/color.gif"
image="C:\ArcIMS\WebSite\color.gif" antialiasing="false"/>
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When layer rendering is included in SERVICEINFO and RASTERFILLSYMBOL is restricted.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <CAPABILITIES forbidden="GET_EXTRACT"/>
        <SCREEN dpi="120"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-141.003005981445" miny="29.9125167103556"
maxx="-52.6202812194824" maxy="83.1083221435546" name="Initial_Extent"
/>
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="Provinces"
id="Provinces">
        <FCLASS type="polygon"> </FCLASS>
        <SIMPLERENDERER>
          <RASTERFILLSYMBOL
url="http://mymachine.domain.com/website/color.gif" transparency="0.5"
overlap="true" />
        </SIMPLERENDERER>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

RASTERMARKERSYMBOL

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Feature ArcMap
Parent elements: **EXACT** **LINE** **OBJECT** **OTHER** **POINT** **POLYGON** **RANGE** **SIMPLERENDERER**

<RASTERMARKERSYMBOL

When using ArcMap Server:
url ="url string"
image ="path to image file"
size ="1,1 - N,N"

When using Image or Feature Server:
url ="url string"
antialiasing ="true | false" **[false]**
hotspot ="0,0 - N,N" **[centered]**
image ="path to image file"
overlap ="true | false" **[true]**
shadow ="0,0,0 - 255,255,255"
size ="1,1 - N,N"
transparency ="0.0 - 1.0" **[1.0]**
usecentroid ="true | false" **[true]**
>

No Child Elements
</RASTERMARKERSYMBOL >

Bold: Attribute or child element is required.

Description:

Symbolizes point features using the specified raster image.

Restrictions:

- *Overlap* is valid only for Image Services. It is ignored when using Feature Services.
- When using an ArcIMS HTML Viewer, ArcIMS Java Viewer, ArcExplorer 4, or any other browser using the ArcIMS Servlet Connector, RASTERFILLSYMBOL is restricted in a SERVICEINFO response. The attribute *image* is not returned. This restriction can be lifted by setting the property *spatialServer.AllowResponsePath* to "true" in esrimap_prop. This property file is found in the servlet engine directory. For more information on the location of esrimap_prop and its properties, see the ArcIMS Help.

This restriction applies only when the ArcIMS Servlet Connector is used. It does not apply to the ActiveX, ColdFusion, or Java Connectors.

- In ArcMap Image Services, symbol is valid only in acetate layers.

Notes:

- In the legend of ArcIMS Java Viewers and ArcExplorer 4, the swatch showing a symbol is limited in size. Images greater than approximately 16x16 pixels in size are truncated to fill the swatch. The full image displays on the map. In the ArcIMS HTML Viewer and other HTML implementations, the size of the swatch can be controlled using LEGEND.
- Acceptable image formats are JPG and GIF.

Attribute Descriptions for RASTERMARKERSYMBOL:

Attribute	Usage
antialiasing	Used to make edges of labels and symbols smoother. When set to "true", antialiasing is active. Note that the time to generate a map may significantly increase when antialiasing is turned on.
hotspot	Determines where marker symbol is placed in relation to actual x,y location of the point the marker symbol represents. A hotspot of 0,0 places the point at the top left corner of the marker symbol. X,y coordinates are positive and measured in pixels. The default hotspot centers the marker symbol over the point based on its actual size. For example, if a marker symbol is 16x16 pixels, the default location is 8,8. If the size attribute is set to 32x32, the default hotspot center is still 8,8.
image	Full pathname to image. ArcIMS Spatial Server uses this pathname to find the image and add it to the map. UNC pathnames can be used (\\myComputer\arcims\output).
overlap	Determines if labels can overlap this symbol. When "true", labels can overlap. When "false", labels will not overlap the symbol. If labels are not drawing as expected, check if *overlap* is set to "false" for this symbol or any other symbol in the ArcIMS service.
shadow	Shadow color using RGB values.
size	Resizes marker symbol to new size in pixels. The default size is the actual width and height of the marker symbol. If *size="0,0"* is specified, the ImageServer writes a warning message to the log file and uses the default size settings.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.
url	URL used by client to retrieve image.
usecentroid	By default, a marker symbol used on polygon layers draws markers at all polygon vertices. If *usecentroid* is "true", marker is placed in the centroid of the polygon. If multiple polygon parts exist, the marker falls on the part with the biggest area.

Examples for RASTERMARKERSYMBOL:

Example 1: When in CONFIG or REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CITIES" visible="true" id="2">
        <DATASET name="CITIES" type="point" workspace="shp_ws-10" />
        <SIMPLERENDERER>
          <RASTERMARKERSYMBOL shadow="0,0,0" overlap="true"
url="http://mymachine.domain.com/website/color.gif"
image="C:\ArcIMS\WebSite\color.gif" transparency="1.0" size="16,16"
hotspot="1,1" antialiasing="false" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When using usecentroid with a polygon layer.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="0">
```

```
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL filltype="solid"
fillcolor="127,227,227" />
          </SIMPLERENDERER>
          <SIMPLERENDERER>
            <RASTERMARKERSYMBOL usecentroid="true"
url="http://mymachine.domain.com/website/color.gif"
image="C:\ArcIMS\WebSite\color.gif" size="16,16" />
          </SIMPLERENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 3: When layer rendering is included in SERVICEINFO and RASTERMARKERSYMBOL is restricted.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <CAPABILITIES forbidden="GET_EXTRACT"/>
        <SCREEN dpi="120"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-141.003005981445" miny="29.9125167103556"
maxx="-52.6202812194824" maxy="83.1083221435546" name="Initial_Extent"
/>
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="Cities"
id="Cities">
        <FCLASS type="point"> </FCLASS>
        <SIMPLERENDERER>
          <RASTERMARKERSYMBOL
url="http://mymachine.domain.com/website/city.gif" overlap="true" />
        </SIMPLERENDERER>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

RASTERSHIELDSYMBOL

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Feature
Parent elements: **EXACT** **OTHER** **RANGE** **SIMPLELABELRENDERER**

<RASTERSHIELDSYMBOL

url ="url string"
antialiasing ="true | false" **[false]**
boundary ="true | false" **[false]**
font ="Any system font" **[Arial]**
fontcolor ="0,0,0 - 255,255,255" **[0,0,0]**
fontsize ="1 - NNN" **[12]**
fontstyle ="regular | bold | italic | underline | outline | bolditalic" **[regular]**
image ="path to image file"
labelmode ="full | numericonly" **[numericonly]**
printmode ="titlecaps | allupper | alllower | none" **[none]**
shadow ="0,0,0 - 255,255,255"
textposition ="0,0 - N,N"
transparency ="0.0 - 1.0" **[1.0]**
>

No Child Elements
</RASTERSHIELDSYMBOL >

Bold: Attribute or child element is required.

Description:

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

Restrictions:

- Works only with line features.
- Boundary attribute valid only with Feature Services.
- When using an ArcIMS HTML Viewer, ArcIMS Java Viewer, ArcExplorer 4, or any other browser using the ArcIMS Servlet Connector, RASTERFILLSYMBOL is restricted in a SERVICEINFO response. The attribute *image* is not returned. This restriction can be lifted by setting the property *spatialServer.AllowResponsePath* to "true" in esrimap_prop. This property file is found in the servlet engine directory. For more information on the location of esrimap_prop and its properties, see the ArcIMS Help.

  This restriction applies only when the ArcIMS Servlet Connector is used. It does not apply to the ActiveX, ColdFusion, or Java Connectors.

- Not valid with ArcMap Server.

Notes:

- Acceptable image formats are JPG and GIF.
- The field for text is specified in the associated label renderer element.
- The image needs to be wide enough to support any long text. The image will not automatically resize to accommodate long strings.
- This element is not available from the ArcIMS Author interface.

Attribute Descriptions for RASTERSHIELDSYMBOL:

Attribute	Usage
antialiasing	Used to make edges of labels and symbols smoother. When set to "true", antialiasing is active. Note that the time to generate a map may significantly increase when antialiasing is turned on.
boundary	Draws a boundary around feature. Only valid with Feature Services.
font	Font name. The name is case sensitive. If font name uses "&", use "&" instead. For example, ESRI Transportation & Civic should be written as ESRI Transportation & Civic. For Feature Services, the font must reside on the client machine or else the system default font is used.
fontcolor	Font color using RGB values.
fontsize	Font size.
fontstyle	Font style.
image	Full pathname to image. ArcIMS Spatial Server uses this pathname to find the image and add it to the map. UNC pathnames can be used (\\myComputer\arcims\output).
labelmode	Determines what value is drawn on raster shield. If "full" is used, the entire field value, such as I-80, is displayed. If "numericonly" is used, only numbers within the field are displayed. For example, I-80 is displayed as 80.
printmode	Determines how labels are printed. If "none" is used, no change is made to the label: Welcome to ArcIMS. If "alllower" is used, all letters are lowercase: welcome to arcims. If "allupper" is used, all letters are uppercase: WELCOME TO ARCIMS. If "titlecaps" is used, the first letter of each word in a label is uppercase and everything else is lowercase: Welcome To Arcims.
shadow	Shadow color using RGB values.
textposition	Determines where text is placed in relation to shield image. The coordinate 0,0 is in the top left corner, and x and y are positive and measured in pixels. If attribute is not used, then text is placed in center of shield image.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.

url URL used by client to retrieve image.

Examples for RASTERSHIELDSYMBOL:

Example 1: When in a SIMPLELABELRENDERER in CONFIG or REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.215027" miny="18.924782" maxx="-66.969849"
maxy="71.406647" name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="ROADS" visible="true" id="1">
        <DATASET name="ROADS" type="line" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <SIMPLELABELRENDERER field="ROUTE">
            <RASTERSHIELDSYMBOL transparency="1.0" font="Arial"
fontstyle="bolditalic" fontsize="16" fontcolor="255,255,255"
shadow="125,125,125" printmode="alllower" labelmode="numericonly"
textposition="4,4" antialiasing="true"
url="http://mymachine.domain.com/website/shield.gif"
image="c:\arcims\website\shield.gif" />
          </SIMPLELABELRENDERER>
          <SIMPLERENDERER>
            <SIMPLELINESYMBOL type="solid" width="1" captype="round"
jointype="round" color="255,0,0" />
          </SIMPLERENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When in a VALUEMAPLABELRENDERER in CONFIG or REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
```

```
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.215027" miny="18.924782" maxx="-66.969849"
maxy="71.406647" name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="ROADS2" visible="true" id="100">
        <DATASET name="ROADS" type="line" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <VALUEMAPLABELRENDERER lookupfield="ADMN_CLASS"
labelfield="ROUTE" linelabelposition="placeabove">
          <EXACT value="State Highway" label="State Highway">
            <RASTERSHIELDSYMBOL font="Arial" fontstyle="bold"
fontsize="16" fontcolor="255,255,255" labelmode="numericonly"
url="http://mycomputer.domain.com/website/state.gif"
image="c:\arcims\website\state.gif" />
          </EXACT>
          <EXACT value="US Highway" label="US Highway">
            <RASTERSHIELDSYMBOL font="Arial" fontstyle="bold"
fontsize="16" fontcolor="0,0,0" labelmode="numericonly"
url="http://mycomputer.domain.com/website/us.gif"
image="c:\arcims\website\us.gif" />
          </EXACT>
          <EXACT value="Interstate" label="Interstate">
            <RASTERSHIELDSYMBOL font="Arial" fontstyle="bold"
fontsize="16" fontcolor="255,255,255" labelmode="numericonly"
url="http://mycomputer.domain.com/website/interstate.gif"
image="c:\arcims\website\interstate.gif" />
          </EXACT>
          <OTHER>
            <RASTERSHIELDSYMBOL transparency="1.0" font="Arial"
fontstyle="bolditalic" fontsize="16" fontcolor="255,255,255"
shadow="125,125,125"
url="http://mycomputer.domain.com/website/other.gif"
image="c:\arcims\website\other.gif" />
          </OTHER>
          </VALUEMAPLABELRENDERER>
          <SIMPLERENDERER>
            <SIMPLELINESYMBOL type="solid" width="1" captype="round"
jointype="round" color="33,44,27" />
          </SIMPLERENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 3: When layer rendering is included in SERVICEINFO and RASTERSHIELDSYMBOL is restricted.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-141.003005981445" miny="41.9133186340332"
maxx="-52.6202812194824" maxy="83.1083221435546" name="Initial_Extent"
/>
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="province"
id="0">
        <FCLASS type="polygon"> </FCLASS>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="227,227,27" />
        </SIMPLERENDERER>
      </LAYERINFO>
      <LAYERINFO type="featureclass" visible="true" name="ROADS"
id="1">
        <FCLASS type="line"> </FCLASS>
        <GROUPRENDERER>
          <SIMPLELABELRENDERER field="ROUTE">
            <RASTERSHIELDSYMBOL
url="http://mycomputer.domain.com/website/shield.gif" />
          </SIMPLELABELRENDERER>
          <SIMPLERENDERER>
            <SIMPLELINESYMBOL color="255,0,0" />
          </SIMPLERENDERER>
        </GROUPRENDERER>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

RENAME_METADATA

Used in: REQUEST
Servers: Metadata (Publish)
Parent elements: PUBLISH_METADATA

```
<RENAME_METADATA

    docid ="string"
    newname ="string"
>

    No Child Elements
</RENAME_METADATA >
```

Bold: Attribute or child element is required.

Description:

Changes the name of a metadata document.

Restrictions:

None

Notes:

- See METADATA_ACTION for response.

Attribute Descriptions for RENAME_METADATA:

Attribute	Usage
docid	String used to uniquely identify a document. The client used to publish the metadata is responsible for creating the document ID. This ID is automatically assigned when using ArcCatalog. If another client is used, GET_UUID can be used to request a valid ID. The format for an ID is the following: {HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH} where H is a hexadecimal digit (0-9,a-f,A-F). The ID is limited to 38 characters.
newname	New name that identifies the dataset.

Examples for RENAME_METADATA:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <PUBLISH_METADATA>
      <RENAME_METADATA docid="{B597AB56-8B47-11D5-99D1-
000086460FA0}" newname="World_Countries" />
    </PUBLISH_METADATA>
  </REQUEST>
</ARCXML>
```

REQUEST

Used in: REQUEST
Servers: Image Query Feature Extract Geocode Metadata ArcMap
Parent elements: ARCXML

<REQUEST >
No Attributes

<GET_EXTRACT... /> *[Or]*
<GET_FEATURES... /> *[Or]*
<GET_GEOCODE... /> *[Or]*
<GET_IMAGE... /> *[Or]*
<GET_LAYOUT... /> *[Or]*
<GET_METADATA... /> *[Or]*
<GET_RASTER_INFO... /> *[Or]*
<GET_SERVICE_INFO... /> *[Or]*
<PUBLISH_METADATA... /> *[Or]*

</REQUEST >

Bold: Attribute or child element is required.

Description:

Defines request to be sent to an ArcIMS Spatial Server for processing.

Restrictions:

- Only one child element can be used per request.

Notes:

- GET_SERVICE_INFO response is SERVICEINFO.
- GET_IMAGE response is IMAGE.
- GET_FEATURES response is FEATURES.
- GET_GEOCODE response is GEOCODE.
- GET_EXTRACT response is EXTRACT.
- GET_LAYOUT response is LAYOUT.
- GET_METADATA response is METADATA.
- PUBLISH_METADATA response is METADATA_ACTION.
- GET_RASTER_INFO response is RASTER_INFO.

Examples for REQUEST:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
  <GET_IMAGE>
    <PROPERTIES>
      <LAYERLIST>
        <LAYERDEF id="Cities">
          <SIMPLERENDERER>
            <SIMPLEMARKERSYMBOL size="16" color="0,0,0" />
          </SIMPLERENDERER>
         <SPATIALQUERY>
            <SPATIALFILTER relation="area_intersection">
            <ENVELOPE maxy="60" maxx="60" miny="0" minx="0" />
         </SPATIALFILTER>
         </SPATIALQUERY>
      </LAYERDEF>
      </LAYERLIST>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
   </PROPERTIES>
  </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

RESET

Used in: REQUEST
Servers: Metadata (Publish)
Parent elements: PUBLISH_METADATA

```
<RESET

    connection ="true | false"   [false]
    tables ="true | false"   [false]
>

    <WORKSPACES... />

</RESET >
```

Description:

Resets a connection to an existing workspace in an ArcSDE database or points to another ArcSDE workspace.

Restrictions:

None

Notes:

- See METADATA_ACTION for response.
- The attribute *tables* must be used with caution. When set to "true", all tables in the repository are deleted and recreated.
- When *connection* is set to "true", an ArcSDE workspace must be specified.

Attribute Descriptions for RESET:

Attribute	Usage
connection	When set to "true", an ArcSDE connection should be re-established. An ArcSDE workspace must be specified.
tables	When set to "true", all ArcSDE tables in the repository are deleted and re-created. Use with caution.

Examples for RESET:

Example 1: When reconnecting to the current ArcSDE database or connecting to another ArcSDE database.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <PUBLISH_METADATA>
      <RESET connection="true">
        <WORKSPACES>
          <SDEWORKSPACE name="sde_esri" server="metadata"
instance="port:5150" encrypted="true" password="LXEMUR" />
        </WORKSPACES>
      </RESET>
    </PUBLISH_METADATA>
  </REQUEST>
</ARCXML>
```

Example 2: When tables="true". Use with caution.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <PUBLISH_METADATA>
      <RESET tables="true" />
    </PUBLISH_METADATA>
  </REQUEST>
</ARCXML>
```

RESPONSE

Used in: **RESPONSE**
Servers: Image Query Feature Extract Geocode Metadata ArcMap
Parent elements: **ARCXML**

<RESPONSE >
No Attributes

<EXTRACT... /> *[Or]*
<FEATURES... /> *[Or]*
<GEOCODE... /> *[Or]*
<IMAGE... /> *[Or]*
<LAYOUT... /> *[Or]*
<METADATA... /> *[Or]*
<METADATA_ACTION... /> *[Or]*
<RASTER_INFO... /> *[Or]*
<SERVICEINFO... /> *[Or]*

</RESPONSE >

Bold: Attribute or child element is required.

Description:

Contains results from a request to the ArcIMS Spatial Server.

Restrictions:

- Only one child element is used per response.

Notes:

- SERVICEINFO is the response to GET_SERVICE_INFO.
- IMAGE is the response to GET_IMAGE.
- FEATURES is the response to GET_FEATURES.
- GEOCODE is the response to GET_GEOCODE.
- EXTRACT is the response to GET_EXTRACT.
- LAYOUT response is GET_LAYOUT.
- RASTER_INFO is the response to GET_RASTER_INFO.
- METADATA is the response to GET_METADATA.
- See METADATA_ACTION for paired requests.

Examples for RESPONSE:

Example 1:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <IMAGE>
      <ENVELOPE minx="-118.19793324" miny="34.03441917" maxx="-
118.12940130" maxy="34.08010713" />
      <OUTPUT file="c:\output\world_MYMACHINE2052765.gif"
url="http://mymachine.domain.com/output/world_MYMACHINE2052765.gif" />
      <LEGEND file="c:\output\world_MYMACHINE2052766.gif"
url="http://mymachine.domain.com/output/world_MYMACHINE2052766.gif" />
    </IMAGE>
  </RESPONSE>
</ARCXML>
```

RESPONSE_COLUMN

Used in: CONFIG
Parent elements: ADMIN_TABLE

<RESPONSE_COLUMN

columnname ***="string"***
>

No Child Elements
</RESPONSE_COLUMN >

Bold: Attribute or child element is required.

Description:

Specifies columns in the administration table. The column values are returned as attributes of each METADATA_DATASET element in a response.

Restrictions:

- Fields cannot contain null values.

Notes:

- Care should be taken in using RESPONSE_COLUMN. One of the purposes of an administration table is to have columns that are hidden from the clients. Using RESPONSE_COLUMN makes both the column name and the column values visible to clients.
- Column names added through RESPONSE_COLUMN become "custom attributes" in METADATA_DATASET. For example, if you include RESPONSE_COLUMN *columnname="service_running"*, then in addition to the extisting attributes in METADATA_DATASET, a new attribute named *service_running* will always be returned as part of METADATA_DATASET. Please note the following:
  - If you are using a DTD, you must update METADATA_DATASET to account for these "custom attributes".
  - You cannot use a column name that replicates any existing attribute in METADATA_DATASET.
- Only one column name can be included in *columnname*. For adding multiple columns, use one REPONSE_COLUMN for each column.
- This element is not available from the ArcIMS Author interface.

Attribute Descriptions for RESPONSE_COLUMN:

Attribute	Usage
columnname	Name of column in the administration table. References can be made only to character columns.

Examples for RESPONSE_COLUMN:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <METADATA_CONFIG>
      <WORKSPACES>
        <SDEWORKSPACE name="unique_name" server="server_name"
instance="port:5151" database="optional_database_name" user="user_name"
password="user_password" />
      </WORKSPACES>
      <ADMIN_TABLE tablename="AdminTable" idcolumn="DocUUID"
filter="Approved='Y'" insert="INSERT INTO AdminTable (DocUUID,
Approved) VALUES ('%s', 'N')">
        <RESPONSE_COLUMN columnname="Approved" />
      </ADMIN_TABLE>
      <METADATA_CONTENT validate="true" />
      <TABLE_NAME prefix="imsmetadata" />
    </METADATA_CONFIG>
  </CONFIG>
</ARCXML>
```

RING

Used in: **CONFIG** **REQUEST** **RESPONSE** **MARKUP**
Servers: Image Query Feature Extract ArcMap
Parent elements: **POLYGON**

<RING >
No Attributes

<COORDS... /> *[Or]*
(m) <POINT... /> *[Or]*
(m) <HOLE... />

</RING >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Provides the x,y coordinate locations of a polygon feature.

Restrictions:

- Either COORDS or POINT is required.

Notes:

- For the different options of representing geometry in an acetate layer, see OBJECT.

Examples for RING:

Example 1: When in SPATIALFILTER in CONFIG or REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_IMAGE>
    <PROPERTIES>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      <IMAGESIZE width="643" height="502" />
    </PROPERTIES>
    <LAYER type="featureclass" name="select layer" visible="true"
id="selected">
      <DATASET fromlayer="countries" />
      <SPATIALQUERY>
        <SPATIALFILTER relation="area_intersection">
          <POLYGON>
            <RING>
              <POINT x="83.15605550814075" y="38.07185101549165" />
              <POINT x="111.09942196116728" y="-4.70645066589869" />
```

```
            <POINT x="155.1079549837513" y="-10.38915084069517" />
            <POINT x="139.1079549837513" y="66.38915084069517" />
            <POINT x="83.15605550814075" y="38.07185101549165" />
          </RING>
        </POLYGON>
      </SPATIALFILTER>
    </SPATIALQUERY>
    <SIMPLERENDERER>
      <SIMPLEPOLYGONSYMBOL fillcolor="0,0,0" filltype="cross" />
    </SIMPLERENDERER>
  </LAYER>
</GET_IMAGE>
</REQUEST>
</ARCXML>
```

Example 2: When using RING in an acetate layer.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
/>
        <IMAGESIZE width="800" height="600" />
      </PROPERTIES>
      <LAYER type="acetate" name="acetate1" id="acetate1">
        <OBJECT units="database">
          <SIMPLEPOLYGONSYMBOL fillcolor="0,255,0" />
          <POLYGON>
            <RING>
              <POINT x="83.15605550814075" y="38.07185101549165" />
              <POINT x="111.09942196116728" y="-4.70645066589869" />
              <POINT x="155.1079549837513" y="-10.38915084069517" />
              <POINT x="139.1079549837513" y="66.38915084069517" />
              <POINT x="83.15605550814075" y="38.07185101549165" />
              <HOLE>
                <POINT x="100.15605550814075" y="20.07185101549165" />
                <POINT x="103.09942196116728" y="30.70645066589869" />
                <POINT x="106.1079549837513" y="30.38915084069517" />
                <POINT x="100.15605550814075" y="20.07185101549165" />
              </HOLE>
            </RING>
          </POLYGON>
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 3: When in a FEATURES response.

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
```

```
<RESPONSE>
  <FEATURES>
    <FEATURE>
      <FIELDS CUST_ID="4" NAME="Customer  4"  #SHAPE#="[Geometry]"
#ID#="3" />
      <POLYGON>
        <RING>
          <POINT x="-133.15605550814075" y="78.07185101549165" />
          <POINT x="-131.09942196116728" y="74.70645066589869" />
          <POINT x="-128.1079549837513" y="76.38915084069517" />
          <POINT x="-128.1079549837513" y="76.38915084069517" />
          <POINT x="-133.15605550814075" y="78.07185101549165" />
        </RING>
      </POLYGON>
    </FEATURE>
  <FEATURECOUNT count="1" hasmore="false" />
</FEATURES>
</RESPONSE>
</ARCXML>
```

SCALE

Used in: REQUEST
Servers: ArcMap
Parent elements: DATAFRAME

```
<SCALE

    rf ="double"
    x ="double"
    y ="double"
>

    No Child Elements
</SCALE >
```

Bold: Attribute or child element is required.

Description:

Defines the dataframe scale and center point.

Restrictions:

None

Notes:

None

Attribute Descriptions for SCALE:

Attribute	Usage
rf	Relative factor such as 1:24000. The value for *rf* in this case would be 24000.
x	X-coordinate representing the center of the map.
y	Y-coordinate representing the center of the map.

Examples for SCALE:

Example 1:

```
<ARCXML version="1.1">
<REQUEST>
  <GET_LAYOUT>
    <PROPERTIES>
      <IMAGESIZE width="500" height="700"/>
        <OUTPUT type="jpg"/>
        <ENVELOPE minx="0" miny="0" maxx="11" maxy="8.5" />
      </PROPERTIES>
      <DATAFRAME id="Layers">
        <SCALE rf="77000" y="37.32" x="-121.91" />
      </DATAFRAME>
    </GET_LAYOUT>
  </REQUEST>
</ARCXML>
```

SCALEBAR

Used in: CONFIG REQUEST
Servers: Image ArcMap
Parent elements: CONFIG OBJECT

<SCALEBAR

*When parent element is **CONFIG in a viewer configuration file:***
backcolor ="0,0,0 - 255,255,255" **[192,192,192]**
fontcolor ="0,0,0 - 255,255,255" **[0,0,0]**
mapunits ="decimal_degrees | feet | meters" **[As defined in map configuration file]**
scaleunits ="miles | feet | meters | kilometers" **[feet]**
screenunits ="inches | centimeters" **[inches]**

*When parent element is **OBJECT with ArcMap Server:***
coords ="***double***"
barcolor ="0,0,0 - 255,255,255" **[255, 162, 115]**
barwidth ="*integer*" **[5]**
font ="Any system font" **[Arial]**
fontcolor ="0,0,0 - 255,255,255" **[0,0,0]**
fontsize ="*integer*" **[10]**
fontstyle ="regular | bold | italic | underline | outline | bolditalic" **[regular]**
mapunits ="degrees | meters | feet" **[degrees]**
scaleunits ="miles | feet | meters | kilometers" **[miles]**
screenlength ="*integer*"

*When parent element is **OBJECT with Image Server:***
coords ="***double***"
antialiasing ="true | false" **[false]**
barcolor ="0,0,0 - 255,255,255" **[255, 162, 115]**
bartransparency ="0.0 - 1.0" **[1]**
barwidth ="*integer*" **[5]**
distance ="*double*"
font ="Any system font" **[Arial]**
fontcolor ="0,0,0 - 255,255,255" **[0,0,0]**
fontsize ="*integer*" **[10]**
fontstyle ="regular | bold | italic | underline | outline | bolditalic" **[regular]**
mapunits ="degrees | meters | feet" **[degrees]**
mode ="cartesian | geodesic" **[cartesian]**
outline ="0,0,0 - 255,255,255"
overlap ="true | false" **[true]**
precision ="*integer*" **[0]**
round ="*double*"
scaleunits ="miles | feet | meters | kilometers" **[miles]**
screenlength ="*integer*"

```
    textransparency ="0.0 - 1.0"  [1]
>

    No Child Elements
</SCALEBAR >
```

Bold: Attribute or child element is required.

Description:

SCALEBAR is used two ways. When the parent element is OBJECT, SCALEBAR defines the look and feel of the scale bar in the acetate layer. When the parent element is CONFIG, SCALEBAR defines the look and feel of the scale bar applet in ArcExplorer 4 or the ArcIMS Java Viewers.

Restrictions:

- When parent element is CONFIG, SCALEBAR is used only in viewer configuration files.

Notes:

None

Attribute Descriptions for SCALEBAR:

*When parent element is **CONFIG in a viewer configuration file:***

Attribute	Usage
backcolor	Background color using RGB values.
fontcolor	Font color using RGB values.
mapunits	Scale bar units.
scaleunits	Scale units.
screenunits	Screen units.

*When parent element is **OBJECT with Image Server:***

Attribute	Usage
antialiasing	Used to make edges of labels and symbols smoother. When set to "true", antialiasing is active. Note that the time to generate a map may significantly increase when antialiasing is turned on.
barcolor	Scale bar color using RGB values.
bartransparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.
barwidth	Scale bar width in pixels.
coords	Scale bar placement location. Coordinate pair is separated by white space by default. The separator can be changed by using SEPARATORS. If using pixel coordinates, "0 0" is in the lower left corner of the map viewer area.

distance	Sets the length of the scale bar to always be the distance specified. The distance units are the same as the scaleunits.
font	Font name. The name is case sensitive. If font name uses "&", use "&" instead. For example, ESRI Transportation & Civic should be written as ESRI Transportation & Civic. For Feature Services, the font must reside on the client machine or else the system default font is used.
fontcolor	Font color using RGB values.
fontsize	Font size.
fontstyle	Font style.
mapunits	Data units on map.
mode	Used when the map units are in decimal degrees. When the mode is "geodesic", the Image Server takes into account the position on the globe when calculating the size of the scale bar symbol. When the mode is "cartesian", the Image Server uses the same calculation for the scale bar for all points on the globe. The calculation is made at the equator.
outline	Outline color for text using RGB values.
overlap	Determines if labels can overlap this symbol. When "true", labels can overlap. When "false", labels will not overlap the symbol. If labels are not drawing as expected, check if *overlap* is set to "false" for this symbol or any other symbol in the ArcIMS service.
precision	Number of decimal places.
round	Number of digits to round.
scaleunits	Screen units.
screenlength	Scale bar length in pixels.
textransparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.

Examples for SCALEBAR:

Example 1: When in an acetate layer in a map configuration file or REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      </PROPERTIES>
      <LAYER type="acetate" name="scalebar" visible="true"
id="acetate">
        <OBJECT units="pixel">
          <SCALEBAR fontcolor="0,0,0" coords="250 20"
barcolor="255,255,255" fontsize="12" screenlength="300" barwidth="3"
mapunits="degrees" antialiasing="true" mode="geodesic" />
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 2: When in a viewer configuration file and used to display a scale bar in the Java viewers.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="6803364.430246" miny="1840363.881158"
maxx="6819426.753985" maxy="1851351.050505" name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
      <SHAPEWORKSPACE name="shp_ws-8" directory="<path to data>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="city" visible="true" id="0">
      <DATASET name="4" type="polygon" workspace="shp_ws-8" />
      <SIMPLERENDERER>
        <SIMPLEPOLYGONSYMBOL fillcolor="127,27,127" filltype="solid" />
      </SIMPLERENDERER>
      </LAYER>
    </MAP>
   <SCALEBAR backcolor="192,192,192" fontcolor="0,0,0"
mapunits="decimal_degrees" scaleunits="FEET" screenunits="INCHES" />
  </CONFIG>
</ARCXML>
```

SCALEDEPENDENTRENDERER

Used in: CONFIG REQUEST RESPONSE
Servers: Image Feature
Parent elements: GROUPRENDERER LAYER LAYERDEF LAYERINFO SCALEDEPENDENTRENDERER

```
<SCALEDEPENDENTRENDERER

    lower ="double"  [1:1]
    upper ="double"  [1:infinity]
>

    <GROUPRENDERER... />  [Or]
    <SCALEDEPENDENTRENDERER... />  [Or]
    <SIMPLELABELRENDERER... />  [Or]
    <SIMPLERENDERER... />  [Or]
    <VALUEMAPLABELRENDERER... />  [Or]
    <VALUEMAPRENDERER... />  [Or]

</SCALEDEPENDENTRENDERER >
```

Bold: Attribute or child element is required.

Description:

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

Restrictions:

- Only one child element can be used inside a SCALEDEPENDENTRENDERER.
- Not valid with ArcMap Server.

Notes:

- Scales can be set in ArcXML using a relative scale or by calculating the number of map units per pixel. A relative scale represents the scale in a ratio such as 1:24000. In this example, 1 meter equals 24000 meters, or 1 inch equals 24000 inches. When using relative scale, always use a colon (:) between the two values.

  Map units per pixel refers to the number of meters, feet, or decimal degrees represented by one pixel in a map. To convert from a relative scale to map units per pixel, the size of a pixel must first be calculated. The formula for finding the number of meters in a pixel is 0.0254 / dpi. The value 0.0254 is the number of meters in an inch, and dpi is the dpi set in the ArcIMS service or request. If no dpi is set in the service or request, the dpi is assumed to be 96. As an example of pixel

size, if the dpi is 96, the pixel size is 0.0254 / 96 or 0.000265 m. To convert from a relative scale to map units per pixel:

1. If the scale is in **meters**. To calculate the number of meters per pixel, take the relative scale and multiply by 0.000265. For example, if the relative scale is 1:24000, then the number of meters per pixel is 24000 * 0.000265, or 6.36 meters.
2. If the scale is in **feet**. Do the calculation for meters (#1). Multiply the result by 3.28 (the number of feet in a meter). For example, if the number of meters per pixel is 6.36, the number of feet is 6.36 * 3.28, or 20.86 feet.
3. If the scale is in **decimal degrees**. For these calculations, the earth is assumed to be an exact circle with a circumference of 40030.174 km. One degree is 111.195 km (40030.174/360 degrees), or 111195 meters. To calculate the number of degrees, first do the calculation for meters (#1). Next, divide the result by 111195. For example, if the number of meters per pixel is 6.36, the number of degrees is 6.36 / 111195, or 0.0000571968.

- For more information on using renderers, see Using ArcXML Renderers.

Attribute Descriptions for SCALEDEPENDENTRENDERER:

Attribute	Usage
Lower	Minimum scale to display renderer using a relative scale such as 1:24000. Scale can also be calculated as the number of map units per pixel.
Upper	Maximum scale to display renderer using a relative scale such as 1:24000. Scale can also be calculated as the number of map units per pixel.

Examples for SCALEDEPENDENTRENDERER:

Example 1: Changing road symbology at a scale of 1:20000000.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-158.204086" miny="19.067062" maxx="-67.097816"
maxy="70.319496" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-2" directory="<path to USA
ESRIDATA>" />
```

```
      </WORKSPACES>
      <LAYER type="featureclass" name="ROADS" visible="true" id="0">
        <DATASET name="ROADS" type="line" workspace="shp_ws-2" />
        <GROUPRENDERER>
          <SCALEDEPENDENTRENDERER upper="1:20000000">
            <GROUPRENDERER>
              <SIMPLERENDERER>
                <SIMPLELINESYMBOL type="solid" width="5" color="0,0,0"
/>
              </SIMPLERENDERER>
              <SIMPLERENDERER>
                <SIMPLELINESYMBOL type="solid" width="3"
color="255,0,0" />
              </SIMPLERENDERER>
              <SIMPLERENDERER>
                <SIMPLELINESYMBOL type="solid" width="1"
color="255,255,255" />
              </SIMPLERENDERER>
            </GROUPRENDERER>
          </SCALEDEPENDENTRENDERER>
          <SCALEDEPENDENTRENDERER lower="1:20000000">
            <SIMPLERENDERER>
              <SIMPLELINESYMBOL type="solid" width="1" color="255,0,0"
/>
            </SIMPLERENDERER>
          </SCALEDEPENDENTRENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: Using VALUEMAPRENDERER and VALUEMAPLABELRENDERER inside a SCALEDEPENDENTRENDERER.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-74.066" miny="40.689" maxx="-73.823"
maxy="40.883" name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="path to data" />
      </WORKSPACES>
      <LAYER type="featureclass" name="Roads" visible="true" id="4"
maxscale="1:35000">
```

```
        <DATASET name="nyc_roads" type="line" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <SCALEDEPENDENTRENDERER lower="1:21300" >
            <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
              <EXACT value="Freeway">
                <SIMPLELINESYMBOL type="solid" width="3"
color="255,0,0" />
              </EXACT>
              <EXACT value="Street">
                <SIMPLELINESYMBOL type="solid" width="1"
color="0,0,255" />
              </EXACT>
              <OTHER>
                <SIMPLELINESYMBOL type="solid" width="1"
color="0,255,0" />
              </OTHER>
            </VALUEMAPRENDERER>
          </SCALEDEPENDENTRENDERER>
          <SCALEDEPENDENTRENDERER upper="1:21300">
            <GROUPRENDERER>
              <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
                <EXACT value="Freeway">
                  <SIMPLELINESYMBOL type="solid" width="9"
color="0,0,0" />
                </EXACT>
                <EXACT value="Street">
                  <SIMPLELINESYMBOL type="solid" width="8"
color="255,255,255" />
                </EXACT>
                <OTHER>
                  <SIMPLELINESYMBOL type="solid" width="1"
color="0,0,255" />
                </OTHER>
              </VALUEMAPRENDERER>
              <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
                <EXACT value="Freeway">
                  <SIMPLELINESYMBOL type="solid" width="7"
color="255,0,0" />
                </EXACT>
              </VALUEMAPRENDERER>
              <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
                <EXACT value="Freeway">
                  <SIMPLELINESYMBOL type="solid" width="3"
color="0,0,0" />
                </EXACT>
              </VALUEMAPRENDERER>
              <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
                <EXACT value="Freeway">
                  <SIMPLELINESYMBOL type="solid" width="1"
color="255,255,255" />
                </EXACT>
              </VALUEMAPRENDERER>
            </GROUPRENDERER>
          </SCALEDEPENDENTRENDERER>
          <SCALEDEPENDENTRENDERER upper="1:25000">
            <VALUEMAPLABELRENDERER lookupfield="ROAD_TYPE"
labelfield="NAME" linelabelposition="PlaceOnTop"
```

```
howmanylabels="One_label_per_name">
              <EXACT value="Freeway">
                <TEXTSYMBOL font="Arial" fontsize="14" fontstyle="bold"
fontcolor="0,0,255" glowing="153,153,153"/>
              </EXACT>
              <EXACT value="Street">
                <TEXTSYMBOL font="Arial" fontsize="14" fontstyle="bold"
fontcolor="0,0,0" />
              </EXACT>
              <OTHER>
                <TEXTSYMBOL font="Arial" fontsize="14"
fontstyle="regular" fontcolor="0,0,255" />
              </OTHER>
            </VALUEMAPLABELRENDERER>
          </SCALEDEPENDENTRENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

SCREEN

Used in: **CONFIG** **RESPONSE**
Servers: Image Feature ArcMap
Parent elements: **ENVIRONMENT**

<SCREEN

dpi **="1 - NNN"**
>

No Child Elements
</SCREEN >

Bold: Attribute or child element is required.

Description:

Indicates the dots per inch (dpi) used when calculating scales for scale-dependent elements such as SCALEDEPENDENTRENDERER, LAYER, and OBJECT.

Restrictions:

None

Notes:

- If SCREEN is not included in the map configuration file, the assumed dpi is 96.

Attribute Descriptions for SCREEN:

Attribute	Usage
dpi	Dots per inch. Used for calculating the correct scale thresholds for scale-dependent elements such as SCALEDEPENDENTRENDERER, LAYER, and OBJECT.

Examples for SCREEN:

Example 1: When in a map configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="192"/>
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.21" miny="18.92" maxx="-66.96"
```

```
maxy="71.41" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CITIES" visible="true" id="2">
        <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL type="square" width="5" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When in a SERVICEINFO response.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-71.0718204242754" miny="42.368904975182"
maxx="-71.0475995680561" maxy="42.3869647980717" name="Initial_Extent"
/>
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="Streets"
id="1" maxscale="0.00004703130261735 83">
        <FCLASS type="line"></FCLASS>
        <EXTENSION type="Geocode" >
          <GCSTYLE name="USAddressZ" />
        </EXTENSION>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

SDEWORKSPACE

Used in: **CONFIG** **REQUEST**
Servers: Image Query Feature Extract Geocode Metadata (Publish)
Parent elements: **WORKSPACES**

<SDEWORKSPACE

instance ="***string***"
name ="***string***"
password ="***string***"
server ="***string***"
user ="***string***"
database ="*string*"
encrypted ="true | false" **[false]**
geoindexdir ="*string*" **[Windows: c:\temp and UNIX: /tmp]**
>

No Child Elements
</SDEWORKSPACE >

Bold: Attribute or child element is required.

Description:

Defines an ArcSDE data source.

Restrictions:

- Must refer to an existing ArcSDE data source.
- Not valid with ArcMap Server.

Notes:

- The SDEWORKSPACE can be used to set up an ArcSDE Direct Connect. For more information on Direct Connect and installation instructions for use with ArcIMS, see the Direct_Connect folder in the documentation folder located on the ArcIMS CD-ROM.
- The following table summarizes whether the attributes *server* and *database* are required or optional when using ArcSDE or ArcSDE direct connect. It also shows an example of correct syntax for the attribute *instance*.

Database	Server	Instance	Database
Oracle with ArcSDE	Required	Port:5151	Optional
Oracle with direct connect	Optional	sde:oracle	Optional
SQL Server with ArcSDE	Required	Port:5151	Required
SQL Server with direct connect	Optional	sde:sqlserver	Required
Other Databases	Required	Port:5151	Optional

The next table summarizes whether the attribute *user* is required, the correct syntax for *password*, and whether the password can be encrypted.

Database	User	Password	Can Encrypt Password
Oracle with ArcSDE	Required	MyPassword	Yes
Oracle with direct connect	Required	MyPassword @net8servicename	No
SQL Server with ArcSDE	Required	MyPassword	Yes
SQL Server with direct connect	Required	MyPassword	No
Other Databases	Required	MyPassword	Yes

- Passwords for ArcSDE datasets, by default, are not encrypted. In order to encrypt a password, you need to connect to the ArcSDE instance while in ArcIMS Author or ArcExplorer 4. Since you cannot connect to an ArcSDE direct connect layer using ArcIMS Author or ArcExplorer Java 3, the password cannot be encrypted for ArcSDE direct connect.
- If layers do not appear in a map, double-check that the workspace information is correct including the user name and password.

Attribute Descriptions for SDEWORKSPACE:

Attribute	Usage
database	ArcSDE database.
encrypted	When set to true, the password for the ArcSDE instance is encrypted. If false, the password is not encrypted.
geoindexdir	Directory where geocoding index is built. On Windows, the default directory is the "temp" directory. On UNIX, the default is /tmp.
instance	Port number for ArcSDE instance. For example, if the port for an ArcSDE dataset is 5151, then the *instance* is "port:5151". For ArcSDE direct connect, the correct value is "sde:<database_vendor>". For ArcIMS 4.0, the two accepted values with Direct Connect are "sde:oracle" and "sde:sqlserver".
name	Workspace name; must be unique among all data sources.
password	Password to access the ArcSDE server. For Oracle ArcSDE Direct Connect, the password must be appended with "@<net8 service name>" .
server	ArcSDE server.
user	User name to access the ArcSDE server.

Examples for SDEWORKSPACE:

Example 1: When in CONFIG and REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
       <ENVELOPE minx="-178" miny="18" maxx="-66.9" maxy="71.4"
name="Initial_Extent" />
       <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
       <SDEWORKSPACE name="sde_ws-2" server="zephyr"
instance="port:5150" database="" user="washoe" encrypted="true"
password="LXEMUR" />
      </WORKSPACES>
      <LAYER type="featureclass" name="ZEPHYR.STREETS" visible="true"
id="1">
       <DATASET name="ZEPHYR.STREETS" type="line" workspace="sde_ws-6"
/>
       <SIMPLERENDERER>
         <SIMPLELINESYMBOL type="solid" width="1" color="227,127,227"
/>
       </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When using ArcSDE raster data.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="200" miny="200" maxx="2000" maxy="2000"/>
      </PROPERTIES>
      <WORKSPACES>
        <SDEWORKSPACE name="sde_ws-1" server="zephyr"
instance="port:5100" database="" user="raster" password="XLMRP" />
      </WORKSPACES>
      <LAYER type="image" name="SDERASTER" visible="true">
        <DATASET workspace="sde_ws-1" name="RASTER_TEST_IMAGE" />
```

```
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 3: Using ArcSDE direct connect.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0"
maxy="83.62303200000002" name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
      <SDEWORKSPACE name="sde_ws-0" server="ARCSDENAME"
instance="sde:oracle" database="" user="sde" encrypted="false"
password="my_password@net8servicename" geoindexdir="C:\Temp\" />
      </WORKSPACES>
      <LAYER type="featureclass" name="country" visible="true" id="0">
      <DATASET name="SDE.COUNTRY" type="polygon" workspace="sde_ws-0"
/>
      <SIMPLERENDERER>
        <SIMPLEPOLYGONSYMBOL boundarytransparency="1.0"
filltransparency="1.0" fillcolor="27,27,27" boundarycaptype="round"
boundarycolor="0,0,0" />
      </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

SEARCH_METADATA

Used in: **REQUEST**
Servers: Metadata (Browse)
Parent elements: **GET_METADATA** **SEARCH_METADATA**

<SEARCH_METADATA

foldermask ="1 - 15"
maxresults ="*integer*" **[All records matching search criteria]**
operator ="and | or" **[and]**
sort ="name | relevance | contenttype | local_area | global_area" **[name]**
sort2 ="name | relevance | contenttype | local_area | global_area"
startresult ="*integer*" **[0]**
>

(m) **<DOCUMENTINFO... />** *[And/Or]*
(m) **<ENVELOPE... />** *[And/Or]*
(m) **<FULLTEXT... />** *[And/Or]*
(m) **<SEARCH_METADATA... />** *[And/Or]*
(m) **<SUBSET... />** *[And/Or]*
(m) **<TAGTEXT... />** *[And/Or]*
(m) **<TAGVALUE... />** *[And/Or]*
<UPDATED... /> *[And/Or]*

</SEARCH_METADATA >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Used to specify criteria for selecting documents.

Restrictions:

- At least one occurrence of SEARCH_METADATA, DOCUMENT, SUBSET, ENVELOPE, FULLTEXT, TAGTEXT, or TAGVALUE is required. Each element can be used multiple times.

Notes:

- Response is zero or more occurrances of METADATA_DATASET.
- The attributes *maxresults* and *startresult* work together to return a specified number of records at a time. For example, assume 10 records should be returned at a time. In this case, *maxresult* would be set to "10". For the first pass, records 0 through 9 would be returned. To retrieve these records, *startresult* would be set to "0". In the second pass, records 10 through 19 should be retrieved. To do this, *startresult* should be set to 10. The programmer is responsible for keeping track

and changing the value of *startresult* programmatically. The ArcIMS Spatial Server does not make this calculation.

- SEARCH_METADATA can be used in a nested query inside a parent SEARCH_METADATA element.

Attribute Descriptions for SEARCH_METADATA:

Attribute	Usage
foldermask	Specifies folder types to be returned in the response. The value of *foldermask* is an integer storing the sum of one or more of the following values: • 1 = Root document • 2 = "Normal" folder • 4 = Document • 8 = Related Services folder For example, use "4" to return documents only. Use "15" (1 + 2 + 4 + 8) to return all types of folders. The different values are: • 1 = Root document only (1) • 2 = "Normal" folders only (2) • 3 = Root document and "normal" folders (1 + 2) • 4 = Documents only (4) • 5 = Root document and documents (1 + 4) • 6 = "Normal" folders and documents (2 + 4) • 7 = Root document, "normal" folders, and documents (1 + 2 + 4) • 8 = Related Services folder only (8) • 9 = Root document and Related Services folder (1 + 8) • 10 = "Normal" folders and Related Services folder (2 + 8) • 11 = Root document, "normal" folders, and Related Services folder (1 + 2 + 8) • 12 = Documents and Related Services folder (4 + 8) • 13 = Root document, documents, and related Services folder (1 + 4 + 8) • 14 = "Normal" folders, documents, and related Services folder (2 + 4 + 8) • 15 = Root document, "normal" folders, documents, and Related Services folder (1 + 2 + 4 + 8)
maxresults	By default, all records meeting the search criteria are returned. This attribute limits the number of returned records to the maximum value assigned to the attribute.
operator	Used to define the operator for the SEARCH_METADATA query.
sort	Preference for ordering results. "Name" orders the results

	alphabetically. "Relevance" lists results from highest to lowest relevance. "Contenttype" sorts and groups results by content type. "Local_area" lists results by area in ascending order. "Global_area" lists results by area in descending order.
sort2	Sorts search results that were batched using *startresult* and *maxresults*. "Name" orders the results alphabetically. "Relevance" lists results from highest to lowest relevance. "Contenttype" sorts and groups results by content type. "Local_area" lists results by area in ascending order. "Global_area" lists results by area in descending order.
startresult	By default, all records meeting the search criteria are returned starting with record 0. This attribute allows a specified record as the start record.

Examples for SEARCH_METADATA:

Example 1: When searching by name or owner of documents using DOCUMENTINFO.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <SEARCH_METADATA>
        <DOCUMENTINFO name="World" />
      </SEARCH_METADATA>
    </GET_METADATA>
  </REQUEST>
</ARCXML>
```

Example 2: When using SUBSET.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <SEARCH_METADATA>
        <SUBSET type="children" docid="{F7DDF21-BC01-4C20-8AA5-
243B33ED0B1E}" />
     </SEARCH_METADATA>
   </GET_METADATA>
  </REQUEST>
</ARCXML>
```

Example 3: When using TAGVALUE.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <SEARCH_METADATA operator="and" maxresults="10"
```

```
startresult="0" >
        <TAGVALUE tag="metadata/idinfo/citation/citeinfo/pubdate"
greaterthanorequalto="20001006" lessthanorequalto="20010104"/>
      </SEARCH_METADATA>
    </GET_METADATA>
  </REQUEST>
</ARCXML>
```

Example 4: When using ENVELOPE, TAGVALUE, FULLTEXT, and SEARCH_METADATA recursively with TAGTEXT.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <SEARCH_METADATA operator="and">
        <ENVELOPE minx="-176.98" miny="18.93" maxx="-66.97"
maxy="71.35" spatialoperator="within" />
        <TAGVALUE
tag="metadata/dataqual/lineage/srcinfo/srccite/citeinfo/pubdate"
greaterthan="1990" />
        <FULLTEXT word="cities"/>
        <SEARCH_METADATA operator="or">
          <TAGTEXT tag="metadata/idinfo/citation/citeinfo/geoform"
word="data"/>
          <TAGTEXT tag="metadata/idinfo/citation/citeinfo/geoform"
word="digital"/>
        </SEARCH_METADATA>
      </SEARCH_METADATA>
    </GET_METADATA>
  </REQUEST>
</ARCXML>
```

SEMANTIC_PAIR

Used in: REQUEST
Servers: Metadata (Publish)
Parent elements: PUT_METADATA_SEMANTIC

```
<SEMANTIC_PAIR

    tag ="string"
    zcode ="string"
    attribute ="string"
>

    No Child Elements
</SEMANTIC_PAIR >
```

Bold: Attribute or child element is required.

Description:

Establishes a relationship between a tag and the numeric code defining semantics from the GEO Attribute Set.

Restrictions:

None

Notes:

- See METADATA_ACTION for response.

Attribute Descriptions for SEMANTIC_PAIR:

Attribute	Usage
attribute	Name of an attribute for the element.
tag	Defines the location of an element inside an XML tree. For example, the element "title" might be inside an XML structure such as <idinfo> <citation> <citeinfo> **<title>...</title>** </citeinfo> </citation> </idinfo> To define "title", the string value for this attribute would be "idinfo/citation/citeinfo/**title**".
zcode	Numeric code defining semantics from the GEO Attribute Set.

Examples for SEMANTIC_PAIR:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <PUBLISH_METADATA>
      <PUT_METADATA_SEMANTIC>
        <SEMANTIC_PAIR tag="idinfo/citation/citeinfo/title" zcode="4"
/>
      </PUT_METADATA_SEMANTIC>
    </PUBLISH_METADATA>
  </REQUEST>
</ARCXML>
```

SEPARATORS

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Query Feature Extract ArcMap
Parent elements: **ENVIRONMENT**

```
<SEPARATORS

    cs ="string"  [white space (" ")]
    ts ="string"  [semicolon (";")]
>

    No Child Elements
</SEPARATORS >
```

Description:

Identifies separators used between x,y coordinates and coordinate pairs and as separators for lists of strings.

Restrictions:

- Separators are limited to one UNICODE character in length.

Notes:

- SEPARATORS only needs to be included when overriding the defaults: a space for the coordinate separator (cs) and a semicolon for the tuple separator (ts).
- Coordinate and tuple separators are used to separate x,y coordinates and coordinate pairs, respectively, in the COORDS element and with the *coords* attribute in POLYGON and LINE.
- A coordinate separator is used with the *coords* attribute in NORTHARROW, POINT, SCALEBAR, and TEXT.
- The tuple separator is used to separate lists of strings in EXACT.

Attribute Descriptions for SEPARATORS:

Attribute	Usage
cs	Coordinate separator is used to separate an x-coordinate from a y-coordinate.
ts	Tuple separator is used to separate coordinate pairs and string lists.

Examples for SEPARATORS:

Example 1: When in CONFIG.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SEPARATORS cs="#" ts="*" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-141.003006" miny="41.913319" maxx="-52.620281"
maxy="83.108322" name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
      <SHAPEWORKSPACE name="shp_ws-16" directory="<path to CANADA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="province" visible="true" id="0">
      <DATASET name="province" type="polygon" workspace="shp_ws-16" />
        <SPATIALQUERY>
          <SPATIALFILTER relation="area_intersection">
            <POLYGON>
              <RING>
                 <COORDS> -92#45*-92#55*-66#55*-54#63*-92#45 </COORDS>
              </RING>
           </POLYGON>
         </SPATIALFILTER>
       </SPATIALQUERY>
      <SIMPLERENDERER>
        <SIMPLEPOLYGONSYMBOL fillcolor="227,127,227" filltype="solid"
/>
      </SIMPLERENDERER>
      </LAYER>
      <LAYER type="acetate" name="Text" id="acetate">
        <OBJECT units="pixel">
          <TEXT coords="200#200" label="Text goes here">
            <TEXTMARKERSYMBOL fontstyle="bold" fontsize="18"
font="Times New Roman" fontcolor="0,255,0" blockout="120,255,0"
glowing="125,125,125" shadow="0,0,0" transparency="0.8"
antialiasing="false" interval="2" overlap="false" printmode="titlecaps"
halignment="right" valignment="center" angle="300" />
          </TEXT>
        </OBJECT>
      </LAYER>
      <LAYER type="acetate" name="Box">
        <OBJECT units="pixel">
          <LINE coords="10#10*400#10*400#80*10#80*10#10">
            <SIMPLELINESYMBOL color="0,0,0" />
          </LINE>
        </OBJECT>
      </LAYER>
    </MAP>
```

```
  </CONFIG>
</ARCXML>
```

Example 2: When in GET_IMAGE request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <ENVIRONMENT>
        <SEPARATORS cs="*" ts="@" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-148.0" miny="35.0" maxx="-46.0" maxy="90.0" />
        <IMAGESIZE width="800" height="600" />
      </PROPERTIES>
      <LAYER type="featureclass" name="new_CNTRY94" id="333">
        <DATASET fromlayer="1" />
        <SPATIALQUERY>
          <SPATIALFILTER relation="area_intersection">
            <POLYGON>
              <RING>
                 <COORDS> -70*45@-50*45@-52*80@-65*80@-70*45  </COORDS>
              </RING>
           </POLYGON>
         </SPATIALFILTER>
       </SPATIALQUERY>
       <SIMPLERENDERER>
         <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="255,0,0" />
       </SIMPLERENDERER>
      </LAYER>
      <LAYER type="acetate" name="Text" id="444">
        <OBJECT units="pixel">
          <TEXT size="32" coords="10*40" label="Text goes here">
            <TEXTMARKERSYMBOL fontstyle="bold" fontsize="32"
font="Arial" fontcolor="0,0,0" glowing="255,255,0" />
          </TEXT>
        </OBJECT>
      </LAYER>
      <LAYER type="acetate" name="Box" id="555">
        <OBJECT units="pixel">
           <LINE coords="10*40@400*40@400*130@10*130@10*40">
           <SIMPLELINESYMBOL color="0,255,0" width="3" />
           </LINE>
        </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

Example 3: When in GET_FEATURES request.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
```

```
    <GET_FEATURES featurelimit="25" beginrecord="0" outputmode="xml"
geometry="false" envelope="true" compact="true">
      <ENVIRONMENT>
        <SEPARATORS cs=" " ts=";" />
      </ENVIRONMENT>
      <LAYER id="1" />
      <SPATIALQUERY subfields="NAME">
          <SPATIALFILTER relation="area_intersection">
            <POLYGON>
              <RING>
                 <COORDS> -92 5;-92 63;-54 63;-54 45;-92 5 </COORDS>
              </RING>
            </POLYGON>
          </SPATIALFILTER>
       </SPATIALQUERY>
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

Example 4: When in a SERVICEINFO response.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="Cities"
id="0">
        <FCLASS type="point">
          <ENVELOPE minx="-165.270004272461" miny="-53.1500015258789"
maxx="177.130187988281" maxy="78.1999969482422" />
        </FCLASS>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

SERVICE

Used in: Application Server RESPONSE Application Server Administration
Parent elements: SERVICES

<SERVICE

*When parent element is **SERVICES in ADMINCMD while adding a Feature Service:***
axl ="***string***"
name ="***string***"
type ="**add**"
vsname ="***string***"

*When parent element is **SERVICES in ADMINCMD while adding an Image or ArcMap Image Service:***
axl ="***string***"
cleanup ="***integer***"
imgloc ="***string***"
imgurl ="***string***"
vsname ="***string***"
imgtype ="JPG | PNG8 | PNG | GIF" **[JPG]**

*When parent element is **SERVICES in ADMINCMD while adding an Image Service:***
imagememory ="**1-NNN**"
name ="***string***"
type ="**add**"

*When parent element is **SERVICES in ADMINCMD while removing, starting, or stopping an ArcIMS service:***
name ="***string***"
type ="**remove | start | stop**"

*When parent element is **SERVICES in Application Server RESPONSE:***
access ="**PUBLIC | PRIVATE**"
name ="***string***"
servicegroup ="***string***"
status ="**ENABLED | DISABLED**"
type ="***string***"
>

*When parent element is **SERVICES in ADMINCMD:***
No Child Elements

*When parent element is **SERVICES in Application Server RESPONSE:***
<CLEANUP... />
<ENVIRONMENT... />

</SERVICE >

Bold: Attribute or child element is required.

Description:

- In an application server RESPONSE, SERVICE returns information for each ArcIMS service on the specified host or server.
- When used in application server administration, SERVICE contains the information needed to start a service from a command line.

Restrictions:

None

Notes:

- SERVICE inside of ADMINCMD is used to administer ArcIMS services from the command line. To do this, two files are used:
  - An ADMINCMD XML file.
  - A batch file or script.

  The ADMINCMD XML file contains the instructions for adding, starting, stopping, and removing services. See the examples for proper construction of an ADMINCMD XML file.

  On Windows, the batch file contains one line:

  java com.esri.aims.admincore.cmd.Exec **http://mymachine.domain.com Username Password** file **filename**

  Where:

  - **http://mymachine.domain.com** is the host machine.
  - **Username** is the user name used to log into ArcIMS Administrator or Manager.
  - **Password** is the password used to log into ArcIMS Administrator or Manager.
  - **Filename** is the full pathname and name of the ADMINCMD XML file, for example, c:\arcims\axl\admincmd.xml.
  - Note: the parameter "file" must be included before the filename.

  The above command can also be typed on the command line in lieu of using the batch file.
  On UNIX, a script file is used. An example script is:
  #!/bin/csh

setenv JARHOME $AIMSHOME/Manager/lib
setenv AIMSHOST **$argv[1]**

java -cp
$JARHOME/jaxp.jar:$JARHOME/parser.jar:$JARHOME/esri_mo10.jar:
$JARHOME/esri_mo10res.jar:$JARHOME/arcims_admincore.jar:$JARHOME/arcims_admin.jar:
$JARHOME/jcert.jar:$JARHOME/jnet.jar:$JARHOME/jsse.jar:$JARHOME/arcims_resadmin.jar com.esri.aims.admincore.cmd.Exec http://$AIMSHOST admin admin file **$argv[2]**

Where:

- **$argv[1]** is the hostname.
- **$argv[2]** is the name of the ADMINCMD XML file, for example, admincmd.xml.
- Note: the parameter "file" must be included before $argv[2].

Attribute Descriptions for SERVICE:

*When parent element is **SERVICES in ADMINCMD while adding a Feature Service:***

Attribute	Usage
axl	Full pathname to map configuration file.
name	Name of ArcIMS service. Service names are case sensitive.
type	Type of action on service.
vsname	Virtual Server name for assigning ArcIMS service.

*When parent element is **SERVICES in ADMINCMD while adding an Image or ArcMap Image Service:***

Attribute	Usage
axl	Full pathname to map configuration file or ArcMap document.
cleanup	Interval for the number of minutes between the deletion of image files from the Output directory.
imgloc	Full pathname to Output directory for generated map and legend images.
imgtype	Image format for the output map images.
imgurl	URL to Output directory for generated map and legend images.
vsname	Virtual Server name for assigning ArcIMS service.

*When parent element is **SERVICES in ADMINCMD while adding an Image Service:***

Attribute	Usage
imagememory	Maximum size of map image allowed in megabytes. A one megabyte

	image is approximately 512 x 512 pixels, or 262144 pixels total.
name	Name of ArcIMS service. Service names are case sensitive.
type	Type of action on service.

*When parent element is **SERVICES in ADMINCMD while removing, starting, or stopping an ArcIMS service:***

Attribute	Usage
name	Name of ArcIMS service. Service names are case sensitive.
type	Type of action on service.

*When parent element is **SERVICES in Application Server RESPONSE:***

Attribute	Usage
access	Access type of the Virtual Server. PUBLIC means that the Virtual Server and ArcIMS service can be accessed directly by requests from a client. By default, Image, Feature, ArcMap, and Metadata Servers are public. When access is PRIVATE, Virtual Servers are accessed by redirecting requests from a public server to a private one. Private virtual servers include Geocode, Query, and Extract Servers.
name	Name of service running on host machine.
servicegroup	Virtual Server name to which the specified ArcIMS service belongs.
status	Identifies if service is currently running or not.
type	Virtual Server type to which the specified ArcIMS service belongs.

Examples for SERVICE:

Example 1: An ADMINCMD XML file for adding and starting an ArcIMS service from the command line.

```
<?xml version="1.0"?>
<ADMINCMD version="1.0">
  <SERVICES>
    <SERVICE type="add" name="world_image"
          axl="c:\arcims\axl\world.axl"
          vsname="ImageServer1"
          imgloc="c:\arcims\output"
          imgurl="http://mycomputer.domain.com/output"
          imagememory="4"
          cleanup="20"
          imgtype="JPG" />

    <SERVICE type="start" name="world_image" />

    <SERVICE type="add" name="world_arcmap"
          axl="c:\arcims\axl\world.mxd"
          vsname="ImageServerArcMap1"
          imgloc="c:\arcims\output"
          imgurl="http://mycomputer.domain.com/output"
```

```
          imagememory="4"
          cleanup="20"
          imgtype="JPG" />

    <SERVICE type="start" name="world_arcmap" />

    <SERVICE type="add" name="world_feature"
          axl="c:\arcims\axl\world.axl"
          vsname="FeatureServer1"
/>

    <SERVICE type="start" name="world_feature" />

  </SERVICES>
</ADMINCMD>
```

Example 2: An ADMINCMD XML file for stopping and deleting an ArcIMS service from the command line.

```
<?xml version="1.0"?>
<ADMINCMD version="1.0">
  <SERVICES>
    <SERVICE type="stop" name="world_image" />
    <SERVICE type="remove" name="world_image" />
  </SERVICES>
</ADMINCMD>
```

Example 3: When in an application server RESPONSE to GETCLIENTSERVICES.

```
<?xml version="1.0"?>
<ARCXML version="1.1">
<RESPONSE>
  <SERVICES>
    <SERVICE name="world_feature" servicegroup="FeatureServer1"
access="PUBLIC" type="FeatureServer" desc=""  status="ENABLED" >
      <ENVIRONMENT>
        <LOCALE country="US" language="en" />
        <UIFONT name="Arial" />
      </ENVIRONMENT>
      <CLEANUP interval="20" />
    </SERVICE>
    <SERVICE name="world_image" servicegroup="ImageServer1"
access="PUBLIC" type="ImageServer" desc=""  status="ENABLED" >
      <ENVIRONMENT>
        <LOCALE country="US" language="en" />
        <UIFONT name="Arial" />
      </ENVIRONMENT>
      <CLEANUP interval="20" />
    </SERVICE>
  </SERVICES>
</RESPONSE>
</ARCXML>
```

SERVICEINFO

Used in: RESPONSE
Servers: Image Query Feature Extract Geocode ArcMap
Parent elements: RESPONSE

<SERVICEINFO >
No Attributes

*When parent element is **RESPONSE for an ArcMap Image Service and the dataframe attribute is not used in GET_SERVICE_INFO:***
(m) <LAYERINFO... />
<ENVIRONMENT... />
<LAYOUTINFO... />
<PROPERTIES... />

*When parent element is **RESPONSE for an ArcMap Image Service and the dataframe attribute is used in GET_SERVICE_INFO:***
(m) <DATAFRAMEINFO... />
<ENVIRONMENT... />
<LAYOUTINFO... />

*When parent element is **RESPONSE for an Image or Feature Service:***
(m) <LAYERINFO... />
<ENVIRONMENT... />
<PROPERTIES... />

</SERVICEINFO >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Provides details about each layer in an ArcIMS service. If the layer data source is an image such as TIFF, only minimum and maximum extent is returned in the response. If the layer is a vector data source, additional information is returned such as the type of feature, feature geometry, rendering and symbolization, and extension parameters.

Restrictions:

- When using an ArcIMS HTML Viewer, ArcIMS Java Viewer, ArcExplorer 4, or any other browser using the ArcIMS Servlet Connector, several elements in a SERVICEINFO response are restricted.
    - In RASTERMARKERSYMBOL, RASTERFILLSYMBOL, and RASTERSHIELDSYMBOL, the attribute *image* is not returned.
    - OUTPUT is not included in the SERVICEINFO response, even if it is included in the map configuration file.

  These restrictions can be lifted by setting the property *spatialServer.AllowResponsePath* to true in esrimap_prop. This property file is found in the servlet engine directory. For more information on the location of Esrimap_prop and its properties, see the ArcIMS Help.

  These restrictions apply only when the ArcIMS Servlet Connector is used. They do not apply to the ActiveX, ColdFusion, or Java Connectors.

- Not valid with Metadata Server.

Notes:

- See GET_SERVICE_INFO for request.
- SERVICEINFO element contains a LAYERINFO child element for each layer defined in the ArcIMS service.
- All layer fields are described in FCLASS child element including the layer's envelope and all available fields.
- If any extensions were defined for a LAYER in a map configuration file, then information about all these extensions is also returned in EXTENSION elements.
- For more details on using GET_SERVICE_INFO and SERVICEINFO including additional examples, see Using GET_SERVICE_INFO and SERVICEINFO with Image and Feature Services and Using GET_SERVICE_INFO and SERVICEINFO with ArcMap Image Services.

Examples for SERVICEINFO:

Example 1: When all GET_SERVICE_INFO attributes are set to true. The Countries layer includes extensions. An acetate layer is also included.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
```

```
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="image" name="Background" visible="true" id="0">
        <ENVELOPE minx="-180" miny="-89.9747543334961"
maxx="179.9423828125" maxy="90" />
      </LAYERINFO>
      <LAYERINFO type="featureclass" visible="true" name="Countries"
id="1">
        <FCLASS type="polygon">
          <ENVELOPE minx="-180" miny="-90" maxx="180"
maxy="83.5960388183594" />
          <FIELD name="AREA" type="8" size="12" precision="3" />
          <FIELD name="NAME" type="12" size="40" precision="0" />
          <FIELD name="ABBREVNAME" type="12" size="12" precision="0" />
          <FIELD name="FIPS_CODE" type="12" size="2" precision="0" />
          <FIELD name="WB_CNTRY" type="12" size="3" precision="0" />
          <FIELD name="#SHAPE#" type="-98" size="0" precision="0" />
          <FIELD name="#ID#" type="-99" size="16" precision="0" />
        </FCLASS>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltransparency="0.0" boundarywidth="2"
/>
        </SIMPLERENDERER>
        <EXTENSION type="extract">
          <EXTRACTPARAMS clip="true">
            <OUTPUTFILE file="world">
              <OUTPUTFIELD name="NAME" alias="Country"/>
            </OUTPUTFILE>
          </EXTRACTPARAMS>
        </EXTENSION>
        <EXTENSION type="Geocode" >
          <GCSTYLE name="SingleField" />
        </EXTENSION>
        <EXTENSION type="StoredQuery">
          <STOREDQUERIES>
            <STOREDQUERY name="Country">
              <QUERY subfields="#SHAPE# AREA NAME ABBREVNAME FIPS_CODE
WB_CNTRY" where="( NAME = [%var%] )" >
              </QUERY>
              <SQVAR name="[%var%]" position="0">
                <FIELD name="NAME" type="12" precision="0" size="40" />
              </SQVAR>
            </STOREDQUERY>
          </STOREDQUERIES>
        </EXTENSION>
      </LAYERINFO>
      <LAYERINFO type="acetate" name="northarrow" visible="true"
id="northarrow"/>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Example 2: When layer rendering is included and RASTERFILLSYMBOL is restricted.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
        <CAPABILITIES forbidden="GET_EXTRACT"/>
        <SCREEN dpi="120"/>
        <IMAGELIMIT pixelcount="1048576" />
      </ENVIRONMENT>
      <PROPERTIES>
        <ENVELOPE minx="-141.003005981445" miny="29.9125167103556"
maxx="-52.6202812194824" maxy="83.1083221435546" name="Initial_Extent"
/>
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" visible="true" name="Provinces"
id="Provinces">
        <FCLASS type="polygon"> </FCLASS>
        <SIMPLERENDERER>
          <RASTERFILLSYMBOL
url="http://mymachine.domain.com/website/color.gif" transparency="0.5"
overlap="true" />
        </SIMPLERENDERER>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Example 3: When request is routed to the Geocode Server.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US" />
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular" />
        <SEPARATORS cs=" " ts=";"/>
      </ENVIRONMENT>
      <LAYERINFO name="Streets" id="4" >
        <EXTENSION type="geocode">
          <GCSTYLE name="USAddressZ" >
            <GCINPUT id="STREET" type="text" label="Street" width="10"
description="street number, street name and type" />
            <GCINPUT id="ZONE" type="text" label="Zone" width="5"
description="zone information" />
            <GCINPUT id="CROSSSTREET" type="text" label="Cross street"
width="10" description="cross street name and type" />
          </GCSTYLE>
        </EXTENSION>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
```

```
</ARCXML>
```

Example 4: When request is sent to an ArcMap Image Service and DATAFRAMEINFO is not included

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US"/>
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular"/>
        <SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576"/>
        <CAPABILITIES forbidden=""/>
      </ENVIRONMENT>
      <LAYOUTINFO pageunits="inches">
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
      </LAYOUTINFO>
      <PROPERTIES>
        <FEATURECOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North_
American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.97869
82]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174532
925199433]]"/>
        <FILTERCOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North_
American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.97869
82]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174532
925199433]]"/>
        <MAPUNITS units="decimal_degrees"/>
        <BACKGROUND color="255,255,255"/>
        <ENVELOPE minx="-178.637433658708" miny="-149.932588181915"
maxx="180.202305702247" maxy="148.932531819555" name="Initial_Extent" />
      </PROPERTIES>
      <LAYERINFO type="image" name="Background" id="1" visible="true">
        <ENVELOPE minx="-180.10415" miny="-89.8896767396583"
maxx="179.900453479317" maxy="90.112625" />
      </LAYERINFO>
      <LAYERINFO type="featureclass" name="Countries" id="0"
visible="true">
        <FCLASS type="polygon">
          <ENVELOPE minx="-180" miny="-90" maxx="180"
maxy="83.5960388183594" />
          <FIELD name="#ID#" type="-99" size="4" precision="0" />
          <FIELD name="#SHAPE#" type="-98" size="0" precision="0" />
          <FIELD name="AREA" type="8" size="12" precision="11" />
          <FIELD name="NAME" type="12" size="40" precision="0" />
          <FIELD name="ABBREVNAME" type="12" size="12" precision="0" />
          <FIELD name="FIPS_CODE" type="12" size="2" precision="0" />
          <FIELD name="WB_CNTRY" type="12" size="3" precision="0" />
        </FCLASS>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
```

```
</ARCXML>
```

Example 5: When request includes the attribute dataframe for ArcMap layouts.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US"/>
        <UIFONT name="Arial" color="0,0,0" size="12" style="regular"/>
        <SEPARATORS cs=" " ts=";"/><SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576"/>
        <CAPABILITIES forbidden=""/>
      </ENVIRONMENT>
      <LAYOUTINFO pageunits="inches">
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
      </LAYOUTINFO>
      <DATAFRAMEINFO name="Layers">
        <PROPERTIES>
          <FEATURECOORDSYS id="4269"/>
          <FILTERCOORDSYS id="4269"/>
          <MAPUNITS units="decimal_degrees"/>
          <BACKGROUND color="255,255,255"/>
          <ENVELOPE minx="-127.714285386824" miny="-6.58527935381"
maxx="-63.9877554315853" maxy="81.1449102179015" name="Initial_Extent"
/>
        </PROPERTIES>
        <LAYERINFO type="featureclass" name="states" id="2"
visible="true">
          <FCLASS type="polygon"></FCLASS>
        </LAYERINFO>
        <LAYERINFO type="featureclass" name="rivers" id="1"
visible="true">
          <FCLASS type="line"></FCLASS>
        </LAYERINFO>
        <LAYERINFO type="featureclass" name="cities" id="0"
visible="true">
          <FCLASS type="point"></FCLASS>
        </LAYERINFO>
      </DATAFRAMEINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

SERVICES

Used in: Application Server RESPONSE Application Server Administration
Parent elements: ADMINCMD Application Server RESPONSE

<SERVICES >
No Attributes

(m) <SERVICE... />

</SERVICES >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Provides details about ArcIMS services in a SERVICES application server RESPONSE or is used to start services from the command line.

Restrictions:

None

Notes:

- See GETCLIENTSERVICES for request.

Examples for SERVICES:

Example 1: When in ADMINCMD.

```
<?xml version="1.0"?>
<ADMINCMD version="1.0">
  <SERVICES>
    <SERVICE type="stop" name="world_image" />
    <SERVICE type="remove" name="world_image" />
  </SERVICES>
</ADMINCMD>
```

Example 2: When in an application server RESPONSE to GETCLIENTSERVICES.

```
<?xml version="1.0"?>
<ARCXML version="1.1">
<RESPONSE>
  <SERVICES>
    <SERVICE name="world_image" servicegroup="ImageServer1"
access="PUBLIC" type="ImageServer" version=""  status="ENABLED" >
      <ENVIRONMENT>
        <LOCALE country="US" language="en" />
        <UIFONT name="Arial" />
```

```
      </ENVIRONMENT>
      <CLEANUP interval="20" />
    </SERVICE>
  </SERVICES>
</RESPONSE>
</ARCXML>
```

SHAPEWORKSPACE

Used in: **CONFIG** **REQUEST**
Servers: Image Query Feature Extract Geocode
Parent elements: **WORKSPACES**

<SHAPEWORKSPACE

directory ***="string"***
name ***="string"***
codepage ="*string*"
geoindexdir ="*string*" **[same as directory with shapefile]**
shared ="true | false" **[false]**
>

No Child Elements
</SHAPEWORKSPACE >

Bold: Attribute or child element is required.

Description:

Defines a shapefile data source.

Restrictions:

- Must refer to an existing data source.
- Not valid with ArcMap Server.

Notes:

None

Attribute Descriptions for SHAPEWORKSPACE:

Attribute	Usage
codepage	Allows the codepage setting in the DBF header to be redefined. The value is the name of the ICU transcoder. Examples include cp1252 and UTF8.
directory	Directory containing shapefiles. UNC pathnames can be used (\\myComputer\shapefiledirectory).
geoindexdir	Directory where geocoding index is built.
name	Workspace name. Must be unique among all data sources.
shared	When set to "true", the ArcIMS Spatial Server checks if a shapefile has been modified outside of ArcIMS. While a shapefile is being edited, ArcIMS sends a message to the ArcIMS Spatial Server log files saying that an update is in progress. When set to "false", the ArcIMS Spatial Server does not check whether a shapefile has been updated. Access to

the shapefile is faster, but the integrity of the shapefile is at risk if it is modified. It is recommended to use "true" unless safeguards are in place to assure that the shapefile is not edited. Note: In general, access to shapefiles is much faster if the shapefiles reside on the same machine as the ArcIMS Spatial Server. If the shapefiles are on a separate machine, access is faster when set to "false", but the integrity of the shapefile is at risk.

Examples for SHAPEWORKSPACE:

Example 1: When in CONFIG and REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.21" miny="18.92" maxx="-66.96"
maxy="71.41" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CITIES" visible="true" id="2">
      <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLEMARKERSYMBOL type="square" width="5" />
      </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

SHIELDSYMBOL

Used in: CONFIG REQUEST RESPONSE
Servers: Image Feature
Parent elements: EXACT OTHER RANGE SIMPLELABELRENDERER

```
<SHIELDSYMBOL

    type ="interstate | usroad | rect | oval"
    antialiasing ="true | false"  [false]
    font ="Any system font"  [Arial]
    fontcolor ="0,0,0 - 255,255,255"  [0,0,0]
    fontsize ="1 - NNN"  [12]
    fontstyle ="regular | bold | italic | underline | outline | bolditalic"  [regular]
    labelmode ="full | numericonly"  [numericonly]
    minsize ="1 - NNN"  [1]
    shadow ="0,0,0 - 255,255,255"
>

    No Child Elements
</SHIELDSYMBOL >
```

Bold: Attribute or child element is required.

Description:

Symbol for drawing a predefined set of highway shields: U.S. Interstate, U.S. Highway, white rectangle, and white oval.

Restrictions:

- Works only with line features.
- Not valid with ArcMap Server.

Notes:

- The field for text is specified in the associated label renderer element.
- *Labelmode="full"* is designed for a maximum of four characters. If more than four characters are needed, RASTERSHIELDSYMBOL should be used.

Attribute Descriptions for SHIELDSYMBOL:

Attribute	Usage
antialiasing	Used to make edges of labels and symbols smoother. When set to "true", antialiasing is active. Note that the time to generate a map may significantly increase when antialiasing is turned on.
font	Font name. The name is case sensitive. If font name uses "&", use "&" instead. For example, ESRI Transportation & Civic should be written as ESRI Transportation & Civic. For Feature Services, the

	font must reside on the client machine or else the system default font is used.
fontcolor	Font color using RGB values.
fontsize	Font size.
fontstyle	Font style.
labelmode	Determines what value is drawn on the shield. If "full" is used, the entire field value, such as I-80, is displayed. If "numericonly" is used, only numbers within the field are displayed. For example, I-80 is displayed as 80.
minsize	Sets shield size to minimum size in characters. By default, shield expands to length of text.
shadow	Shadow color using RGB values.
type	Symbol type.

Examples for SHIELDSYMBOL:

Example 1: When in a SIMPLELABELRENDERER.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.215027" miny="18.924782" maxx="-66.969849"
maxy="71.406647" name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="ROADS" visible="true" id="1">
        <DATASET name="ROADS" type="line" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <SIMPLERENDERER>
            <SIMPLELINESYMBOL transparency="1.0" type="solid" width="8"
captype="round" jointype="round" color="27,127,27" />
          </SIMPLERENDERER>
          <SIMPLELABELRENDERER field="ROUTE"
linelabelposition="placeontop">
            <SHIELDSYMBOL antialiasing="true" font="Arial"
shadow="0,0,0" fontstyle="regular" fontsize="10"
fontcolor="255,255,255" labelmode="numericonly" type="interstate" />
          </SIMPLELABELRENDERER>
        </GROUPRENDERER>
```

```
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When in a VALUEMAPLABELRENDERER.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.215027" miny="18.924782" maxx="-66.969849"
maxy="71.406647" name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="ROADS" visible="true" id="1">
        <DATASET name="ROADS" type="line" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <VALUEMAPLABELRENDERER lookupfield="ADMN_CLASS"
labelfield="ROUTE" linelabelposition="placeabove">
          <EXACT value="State Highway" label="State Highway">
            <SHIELDSYMBOL antialiasing="true" font="Arial"
fontstyle="regular" fontsize="10" type="oval" />
          </EXACT>
          <EXACT value="US Highway" label="US Highway">
            <SHIELDSYMBOL antialiasing="true" font="Arial"
fontstyle="regular" fontsize="10" type="usroad" />
          </EXACT>
          <EXACT value="Interstate" label="Interstate">
            <SHIELDSYMBOL labelmode="numericonly" antialiasing="true"
font="Tahoma" fontstyle="italic" fontsize="14" type="interstate"
minsize="1"/>
          </EXACT>
          </VALUEMAPLABELRENDERER>
          <SIMPLERENDERER>
            <SIMPLELINESYMBOL type="solid" width="1" captype="round"
jointype="round" color="127,127,27" />
          </SIMPLERENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

SIMPLELABELRENDERER

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Feature
Parent elements: **GROUPRENDERER** **LAYER** **LAYERDEF** **LAYERINFO** **SCALEDEPENDENTRENDERER**

<SIMPLELABELRENDERER

field ="*string*"
featureweight ="no_weight | med_weight | high_weight" **[no_weight]**
howmanylabels ="one_label_per_name | one_label_per_shape | one_label_per_part"
labelbufferratio ="*double*"
labelpriorities ="0,0,0,0,0,0,0,0 - 8,8,8,8,8,8,8,8 | LE_PlaceOnTopHorizontal"
[2,2,1,4,5,3,2,4]
labelweight ="no_weight | med_weight | high_weight" **[high_ weight]**
linelabelposition ="See table below for values" **[PlaceAbove]**
rotationalangles ="*string*"
>

<CALLOUTMARKERSYMBOL... /> *[Or]*
<CHARTSYMBOL... /> *[Or]*
<RASTERSHIELDSYMBOL... /> *[Or]*
<SHIELDSYMBOL... /> *[Or]*
<TEXTSYMBOL... /> *[Or]*

</SIMPLELABELRENDERER >

Bold: Attribute or child element is required.

Description:

Used for labeling features. A field is specified for labeling all features of a particular layer.

Restrictions:

- Use only one symbol inside a SIMPLELABELRENDERER.
- The attribute *labelpriorities="LE_PlaceOnTopHorizontal"* is limited to Image Services.
- Not valid with ArcMap Server.

Notes:

- When rotating symbols, the attribute *labelpriorities* always takes precedence over the attribute *rotationalangles*. If you find that your labels are not rotating as expected when using Image Services, remove the *labelpriorities* attribute if it is present.
- If *subfields* is used in SPATIALQUERY or QUERY for a layer, any fields used for labeling must be included in the subfields list.

- For more information on using renderers, see Using ArcXML Renderers.

Attribute Descriptions for SIMPLELABELRENDERER:

Attribute	Usage
featureweight	Prioritizes the importance of features. The feature weight determines how important the feature labeled is for the label placement algorithm. If "no_weight" is specified, then the feature has no importance and can be labeled over. If "high_weight" is specified, then the feature has high importance and cannot be labeled over. Giving importance to features increases the complexity of the labeling problem and also the processing time.
field	Field containing text for labeling features. The field can be in the layer table or in a joined table. • For shapefiles with no joined tables, the field can be referenced using the short format. field="AREA" • For shapefiles with joined tables, the name of the joined table must be included along with the field. field="JOINEDTABLE.AREA" Files with joined fields cannot be read locally in ArcIMS Author or ArcExplorer 4. • For ArcSDE layers without joined tables, the field can be referenced using the short format. field="AREA" The full long name can also be used, but when a long name is used, the file cannot be read locally in ArcIMS Author or ArcExplorer 4. field="ARCSDENAME.TABLE.AREA" • For ArcSDE layers with joined tables, joined fields must be referenced using the full long format. field="ARCSDENAME.TABLE.AREA" Files with joined fields cannot be read locally in ArcIMS Author or ArcExplorer 4. Labels can include data concatenated from two or more fields. Each field name is separated by a space: • Using a shapefile with no joined tables. The short format can be used for field names. field="CITY STATE_NAME" • Using a shapefile with joined tables. The name of the joined table must be included along with the field names. field="JOINEDTABLE.CITY JOINEDTABLE.STATE_NAME" Files with joined fields cannot be read locally in ArcIMS

	Author or ArcExplorer 4. • Using an ArcSDE layer without joined tables. The short format can be used. field="CITY STATE_NAME" The full long name can also be used. field="ARCSDENAME.TABLE.CITY ARCSDENAME.JOINEDTABLE.STATE_NAME" Files with full long names cannot be read locally in ArcIMS Author or ArcExplorer 4. • Using an ArcSDE layer with joined tables. The full long format must be used. field="ARCSDENAME.TABLE.CITY ARCSDENAME.JOINEDTABLE.STATE_NAME" Files with full long names cannot be read locally in ArcIMS Author or ArcExplorer 4.
howmanylabels	Determines how often a feature is labeled. • "One_label_per_part" labels all parts of a feature in the case of multipart features. For instance, the State of Hawaii consists of several islands, but they are grouped together as one polygon feature. Each island is labeled. • "One_label_per_name" labels once per feature name. When several features share the same name, only one label is shown. For example, if there is more than one polygon with the name "Residential", only one "Residential" polygon is labeled. • "One_label_per_shape" labels once per feature even if there are multiple segments. For example, the group of islands that make up Hawaii are only labeled once. If there are other features with the same name, they are also labeled.
labelbufferratio	Used to set a buffer around the label. When this is set, no labels overlap within the buffer range. The ratio is the fraction of the height or the width of the label rectangle (whichever is smaller) compared to the width of the buffer. A ratio of "0.0" means no buffer. A ratio of "1.0" means that the buffer is twice the size of the label (the label width equals the buffer width). A negative ratio causes the buffer to be smaller than the label. This can be used to allow labels to overlap.
labelpriorities	Used to determine where to place the label around the point. The attribute accepts different weights for each of eight positions around the point. Each position corresponds to the positions as shown below: 1 2 3 8 **X** 4 7 6 5

	In each position, the user can prioritize the importance of that position from 0 upwards. 0 signifies that the label should not be placed in that position. 1 means that this is an acceptable position for the label, and all higher numbers represent lesser priorities for that position. For example, "1,0,1,0,0,0,0,0" means that only the first and third label positions will be taken into account when labeling. In another example, "1,2,3,0,1,0,0,0" means try to label at the first and fifth position; if not, then put the label at the second position; if not, then put it at the third position; if this is not possible, then don't label it. <SIMPLELABELRENDERER field="NAME" labelpriorities="1,2,3,0,1,0,0,0"> Another option is to place a label on top of points rather than around points. To do this, use "LE_PlaceOnTopHorizontal" for the label priority. <SIMPLELABELRENDERER field="NAME" labelpriorities="LE_PlaceOnTopHorizontal">
labelweight	Used to prioritize the importance of labels. The label weight is usually set to "high_weight" since the labels are more important. This can be set lower if the labels are not as important as the feature.
linelabelposition	Determines where on the line to place the label. The following options are available: • PlaceAbove - Place above the line. • PlaceBelow - Place below the line. • PlaceOnTop - Place on the line. • PlaceLeft - Place along the left side of the line (Label follows line and is not perpendicular to the line). • PlaceRight - Place along the right side of the line (Label follows line and is not perpendicular to the line). • PlaceAboveBelow - Place above or below the line. • PlaceLeftRight - Place at either side of the line. • PlaceInLine - Place anywhere on the line. • PlaceParallel - Place parallel to the line. • PlaceOnTopHorizontal - Place label on top of the line but always horizontal. • PlaceAtStartAbove - Place label at the start above the line. • PlaceAtStartOnTop - Place label at the start on top of the line. • PlaceAtStartBelow - Place label at the start below the line. • PlaceAtEndAbove - Place label at the end above the line. • PlaceAtEndOnTop - Place label at the end on top of the line. • PlaceAtEndBelow - Place label at the end below the line. • PlaceEitherEndAbove - Place at either end above the line. • PlaceEitherEndOnTop - Place at either end on top of the line.

- PlaceEitherEndBelow - Place at either end below the line.

<VALUEMAPLABELRENDERER lookupfield="ADMN_CLASS" labelfield="ROUTE" linelabelposition="PlaceOnTop">

rotationalangles	The rotational angles are possible angles that the label can be placed at, relative to the labeled point. By default, labels are always placed horizontally. To rotate a label, a comma-delimited list of up to eight rotational angles can be given and are prioritized from first to last. For example, if the first priority is to place labels at 45 degrees and the second priority is at 30 degrees, the rotational angles attribute would look like this: <SIMPLELABELRENDERER field="NAME" rotationalangles="45,30"> *Labelpriorities* always take precedence over *rotationalangles*. If you find that your labels are not rotating as expected, remove the *labelpriorities* attribute if it is present. Alternatively, you can set all the *labelpriorities* to "0". <SIMPLELABELRENDERER field="NAME" rotationalangles="45,30" > or <SIMPLELABELRENDERER field="NAME" labelpriorities="0,0,0,0,0,0,0,0" rotationalangles="45,30" >

Examples for SIMPLELABELRENDERER:

Example 1: When using a point layer.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="cities" visible="true" id="2">
        <DATASET name="cities" type="point" workspace="shp_ws-0" />
```

```
        <GROUPRENDERER>
          <SIMPLELABELRENDERER field="NAME"
labelpriorities="0,0,1,2,2,0,0,0">
            <TEXTSYMBOL font="Arial" fontstyle="regular" fontsize="10"
/>
          </SIMPLELABELRENDERER>
          <SIMPLERENDERER>
            <SIMPLEMARKERSYMBOL color="255,0,255" width="8" />
          </SIMPLERENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When using a line layer.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.215027" miny="18.924782" maxx="-66.969849"
maxy="71.406647" name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="ROADS" visible="true" id="2">
        <DATASET name="ROADS" type="line" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <SIMPLELABELRENDERER field="ROUTE"
linelabelposition="PlaceOnTop" howmanylabels="one_label_per_shape">
            <TEXTSYMBOL antialiasing="true" font="Arial"
fontstyle="regular" fontsize="10" />
          </SIMPLELABELRENDERER>
          <SIMPLERENDERER>
            <SIMPLELINESYMBOL type="solid" width="1" color="127,27,127"
/>
          </SIMPLERENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 3: When using a polygon layer.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="10">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <SIMPLERENDERER >
              <SIMPLEPOLYGONSYMBOL filltype="solid"
fillcolor="113,169,249" />
          </SIMPLERENDERER>
          <SIMPLELABELRENDERER field="NAME"
howmanylabels="one_label_per_part">
            <TEXTSYMBOL antialiasing="true" font="Comic Sans MS"
fontsize="10" />
          </SIMPLELABELRENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

SIMPLELINESYMBOL

Used in: CONFIG REQUEST RESPONSE
Servers: Image Feature ArcMap
Parent elements: EXACT LINE OBJECT OTHER POLYGON RANGE SIMPLERENDERER

<SIMPLELINESYMBOL

When using ArcMap Server:
color ="0,0,0 - 255,255,255" **[0,0,0]**
type ="solid" **[solid]**
width ="0 - NNN" **[0]**

When using Image or Feature Server:
antialiasing ="true | false" **[false]**
captype ="butt | round | square" **[butt]**
color ="0,0,0 - 255,255,255" **[0,0,0]**
jointype ="round | miter | bevel" **[round]**
overlap ="true | false" **[true]**
transparency ="0.0 - 1.0" **[1.0]**
type ="solid | dash | dot | dash_dot | dash_dot_dot" **[solid]**
width ="0 - NNN" **[0]**
>

No Child Elements
</SIMPLELINESYMBOL >

Description:

Symbol for line features.

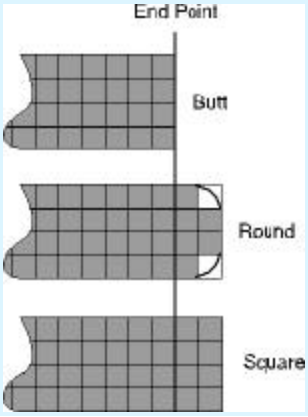
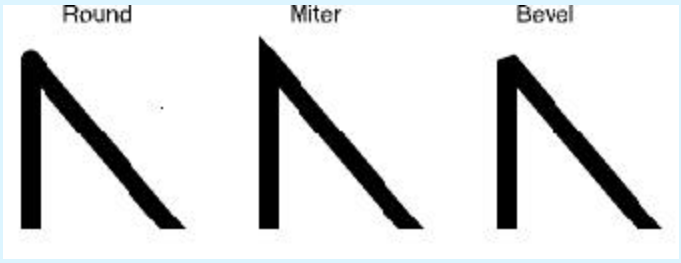
Restrictions:

- *Overlap* is valid only for Image Services. It is ignored when using Feature Services.
- In ArcMap Image Services, symbol is valid only in acetate layers.

Notes:

None

Attribute Descriptions for SIMPLELINESYMBOL:

Attribute	Usage
antialiasing	Used to make edges of labels and symbols smoother. When set to "true", antialiasing is active. Note that the time to generate a map may significantly increase when antialiasing is turned on.
captype	Line end style. 
color	Symbol color using RGB values.
jointype	Line join style. 
overlap	Determines if labels can overlap this symbol. When "true", labels can overlap. When "false", labels will not overlap the symbol. If labels are not drawing as expected, check if *overlap* is set to "false" for this symbol or any other symbol in the ArcIMS service.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.
type	Line type.
width	Line width in pixels.

Examples for SIMPLELINESYMBOL:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.215027" miny="18.924782" maxx="-66.969849"
maxy="71.406647" name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="ROADS" visible="true" id="1">
        <DATASET name="ROADS" type="line" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLELINESYMBOL transparency="0.7" type="dash" width="1"
color="127,227,27" antialiasing="false" overlap="true" captype="square"
jointype="miter" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: Using three line symbols to create a complex line symbol.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.215027" miny="18.924782" maxx="-66.969849"
maxy="71.406647" name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="ROADS" visible="true" id="2">
        <DATASET name="ROADS" type="line" workspace="shp_ws-0" />
```

```
        <GROUPRENDERER>
          <SIMPLERENDERER>
            <SIMPLELINESYMBOL transparency="1.0" type="solid"
width="8" captype="round" jointype="round" color="0,0,0" />
          </SIMPLERENDERER>
          <SIMPLERENDERER>
            <SIMPLELINESYMBOL transparency="1.0" type="solid"
width="6" captype="round" jointype="round" color="255,0,0" />
          </SIMPLERENDERER>
          <SIMPLERENDERER>
            <SIMPLELINESYMBOL transparency="1.0" type="solid"
width="1" captype="round" jointype="round" color="255,255,255" />
          </SIMPLERENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

SIMPLEMARKERSYMBOL

Used in: CONFIG REQUEST RESPONSE
Servers: Image Feature ArcMap
Parent elements: EXACT LINE OBJECT OTHER POINT POLYGON RANGE SIMPLERENDERER

<SIMPLEMARKERSYMBOL

When using ArcMap Server:
color ="0,0,0 - 255,255,255" **[0,0,0]**
outline ="0,0,0 - 255,255,255"
type ="circle | triangle | square | cross | star" **[circle]**
width ="1 - NNN" **[3]**

When using Image or Feature Server:
antialiasing ="true | false" **[false]**
color ="0,0,0 - 255,255,255" **[0,0,0]**
outline ="0,0,0 - 255,255,255"
overlap ="true | false" **[true]**
shadow ="0,0,0 - 255,255,255"
transparency ="0.0 - 1.0" **[1.0]**
type ="circle | triangle | square | cross | star" **[circle]**
usecentroid ="true | false" **[true]**
width ="1 - NNN" **[3]**
>

No Child Elements
</SIMPLEMARKERSYMBOL >

Description:

Symbolizes point features using one of the predefined symbol types: circle, triangle, square, cross, or star.

Restrictions:

- *Overlap* is valid only for Image Services. It is ignored when using Feature Services.
- In ArcMap Image Services, symbol is valid only in acetate layers.
- **Known limitation with *outline* in ArcMap Image Services:** An outline does not show when the attribute *type* is "triangle" or "star".

Notes:

None

Attribute Descriptions for SIMPLEMARKERSYMBOL:

Attribute	Usage
antialiasing	Used to make edges of labels and symbols smoother. When set to "true", antialiasing is active. Note that the time to generate a map may significantly increase when antialiasing is turned on.
color	Symbol color using RGB values.
outline	Outline color using RGB values.
overlap	Determines if labels can overlap this symbol. When "true", labels can overlap. When "false", labels will not overlap the symbol. If labels are not drawing as expected, check if *overlap* is set to "false" for this symbol or any other symbol in the ArcIMS service.
shadow	Shadow color using RGB values.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.
type	Symbol type.
usecentroid	By default, a marker symbol used on polygon layers draws markers at all polygon vertices. If *usecentroid* is "true", marker is placed in the centroid of the polygon. If multiple polygon parts exist, the marker falls on the part with the biggest area.
width	Symbol width in pixels.

Examples for SIMPLEMARKERSYMBOL:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.215027" miny="18.924782" maxx="-66.969849"
maxy="71.406647" name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="cities" visible="true" id="1">
        <DATASET name="cities" type="point" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL transparency="1.0" color="0,255,0"
```

```
type="square" width="16" shadow="0,0,0" outline="255,0,0"
antialiasing="true" overlap="true"/>
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When using usecentroid with a polygon layer.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0"
maxy="83.59603881835938" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-6" directory="<path to WORLD
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="0">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-6" />
        <GROUPRENDERER>
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL fillcolor="127,227,227" />
          </SIMPLERENDERER>
          <SIMPLERENDERER >
            <SIMPLEMARKERSYMBOL usecentroid="true"
color="127,127,227" width="18" />
          </SIMPLERENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

SIMPLEPOLYGONSYMBOL

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Feature ArcMap
Parent elements: **EXACT** **OBJECT** **OTHER** **POLYGON** **RANGE** **SIMPLERENDERER**

<SIMPLEPOLYGONSYMBOL

When using ArcMap Server:
boundary ="true | false" **[true]**
boundarycolor ="0,0,0 - 255,255,255" **[0,0,0]**
boundarytype ="solid" **[solid]**
boundarywidth ="1 - NNN" **[1]**
fillcolor ="0,0,0 - 255,255,255" **[0,200,0]**
fillinterval ="2 - NNN" **[6]**
filltype ="solid | bdiagonal | fdiagonal | cross | diagcross | horizontal | vertical" **[solid]**

When using Image or Feature Server:
antialiasing ="true | false" **[false]**
boundary ="true | false" **[true]**
boundarycaptype ="butt | round | square" **[butt]**
boundarycolor ="0,0,0 - 255,255,255" **[0,0,0]**
boundaryjointype ="round | miter | bevel" **[round]**
boundarytransparency ="0.0 - 1.0" **[1]**
boundarytype ="solid | dash | dot | dash_dot | dash_dot_dot" **[solid]**
boundarywidth ="1 - NNN" **[1]**
fillcolor ="0,0,0 - 255,255,255" **[0,200,0]**
fillinterval ="2 - NNN" **[6]**
filltransparency ="0.0 - 1.0" **[1]**
filltype ="solid | bdiagonal | fdiagonal | cross | diagcross | horizontal | vertical | gray | lightgray | darkgray" **[solid]**
overlap ="true | false" **[true]**
transparency ="0.0 - 1.0" **[no default]**
>

No Child Elements
</SIMPLEPOLYGONSYMBOL >

Description:

Symbol for polygon features.

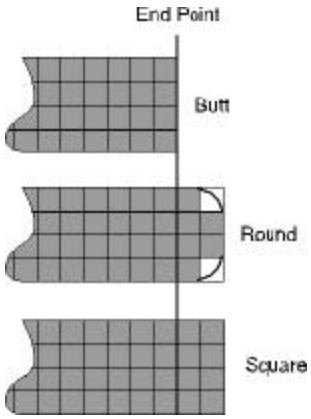
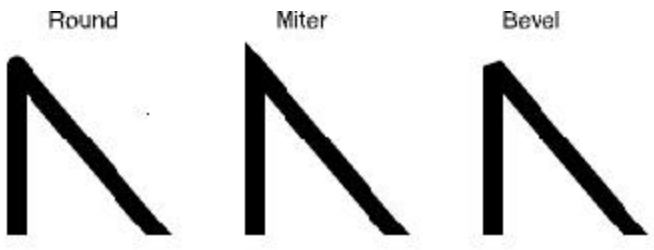
Restrictions:

- *Overlap* is valid only for Image Services. It is ignored when using Feature Services.
- In ArcMap Image Services, symbol is valid only in acetate layers.

Notes:

- For more complex boundary symbols, SIMPLELINESYMBOL can be used on polygon layers.
- *Transparency* takes precedence over *filltransparency* and *boundarytransparency*.
- To make both the fill and boundary of a polygon transparent, set *filltransparency="0.0"* and *boundarytransparency="0.0"*. Do not use *transparency*.

Attribute Descriptions for SIMPLEPOLYGONSYMBOL:

Attribute	Usage
antialiasing	Used to make edges of labels and symbols smoother. When set to "true", antialiasing is active. Note that the time to generate a map may significantly increase when antialiasing is turned on.
boundary	Turns boundary on or off.
boundarycaptype	Boundary end style. 
boundarycolor	Boundary color using RGB values.
boundaryjointype	Boundary join style. 
boundarytransparency	Value to set percentage of transparency for the polygon boundaries. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.

boundarytype	Boundary type.
boundarywidth	Boundary width.
fillcolor	Fill color using RGB values.
fillinterval	Distance between lines for hatch fills.
filltransparency	Value to set percentage of transparency for the polygon fill. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.
filltype	Symbol fill type.
overlap	Determines if labels can overlap this symbol. When "true", labels can overlap. When "false", labels will not overlap the symbol. If labels are not drawing as expected, check if *overlap* is set to "false" for this symbol or any other symbol in the ArcIMS service.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.

Examples for SIMPLEPOLYGONSYMBOL:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="0">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL antialiasing="true"
fillcolor="255,0,0" fillinterval="8" filltype="diagcross"
filltransparency="0.6" boundarywidth="4" boundarycolor="64,64,0"
boundarytype="dash_dot_dot" boundarycaptype="round"
boundaryjointype="bevel" boundarytransparency="0.6" boundary="true"
overlap="true" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

SIMPLERENDERER

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Feature
Parent elements: **GROUPRENDERER** **LAYER** **LAYERDEF** **LAYERINFO** **SCALEDEPENDENTRENDERER**

<SIMPLERENDERER >
No Attributes

<GRADIENTFILLSYMBOL... /> *[Or]*
<HASHLINESYMBOL... /> *[Or]*
<RASTERFILLSYMBOL... /> *[Or]*
<RASTERMARKERSYMBOL... /> *[Or]*
<SIMPLELINESYMBOL... /> *[Or]*
<SIMPLEMARKERSYMBOL... /> *[Or]*
<SIMPLEPOLYGONSYMBOL... /> *[Or]*
<TRUETYPEMARKERSYMBOL... /> *[Or]*

</SIMPLERENDERER >

Bold: Attribute or child element is required.

Description:

Displays features using one symbol.

Restrictions:

- One symbol must be specified. Only one is allowed.
- Not valid with ArcMap Server.

Notes:

- If DATASET layer type is point, then only point symbols can be used. If type is line, then line and point symbols can be used. If type is polygon, then polygon, line, and point symbols can be used.
- For more information on using renderers, see Using ArcXML Renderers.

Examples for SIMPLERENDERER:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.215027" miny="18.924782" maxx="-66.969849"
maxy="71.406647" name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="STATES" visible="true" id="0">
        <DATASET name="STATES" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillinterval="6"
boundarytransparency="1.0" fillcolor="227,227,227" filltype="solid"
boundarytype="solid" boundarywidth="1" boundarycolor="0,0,0" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

SPATIALFILTER

Used in: **CONFIG** **REQUEST**
Servers: Image Query Feature Extract ArcMap
Parent elements: **SPATIALQUERY**

<SPATIALFILTER

relation ="area_intersection | envelope_intersection"
>

<ENVELOPE... /> *[Or]*
<MULTIPOINT... /> *[Or]*
<POLYGON... /> *[Or]*
<POLYLINE... /> *[Or]*
<BUFFER... />

</SPATIALFILTER >

Bold: Attribute or child element is required.

Description:

A spatial filter defines the envelope for a spatial query. The envelope can be a rectangle, point, line, polygon, or buffer.

Restrictions:

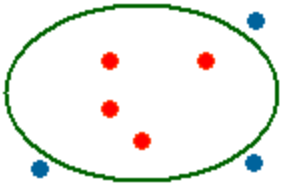
- One ENVELOPE, POLYLINE, POLYGON, or MULTIPOINT element is required. Only one of these elements can be used.
- Using MULTIPOINT as a filter against a point or line layer will yield no results. Instead, a small envelope should be generated around this point. The envelope can then be used as the filter. Similarly, POLYLINE used against a point will yield no results. If desired, an envelope can be generated around the line for use as a filter.

Notes:

- When "area_intersection" is used with *relation*, all features that partially or fully fall within the area of the filter are selected. This results in a complex operation that can take some time. To speed up processing, "envelope_intersection" can be used instead. This method checks to see if the bounding box of the filter overlaps any bounding boxes of the features in the layer. A much quicker search results, but the features found could fall outside the area of the filter. In the examples below, an oval shape is used as the filter. When "envelope_intersection" is used, features outside the filter can be selected even though they are far outside the filter.



Shape of spatial filter to select points



Points selected when relation="area_intersection"



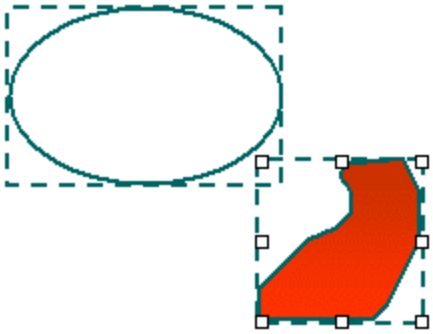
Points selected when relation="envelope_intersection"



Shape of spatial filter to select polygons



No polygons are selected when relation="area_intersection"



Polygon is selected when relation="envelope_intersection"

Attribute Descriptions for SPATIALFILTER:

Attribute	Usage
relation	Describes spatial relation. See Notes section for more details.

Examples for SPATIALFILTER:

Example 1: Note in this example that the DATASET fromlayer is "Countries". This refers to the LAYER id in the map configuration file, not the LAYER name.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_IMAGE>
    <PROPERTIES>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      <IMAGESIZE width="643" height="502" />
    </PROPERTIES>
    <LAYER type="featureclass" name="select layer" visible="true"
id="300">
      <DATASET fromlayer="Countries" />
      <SPATIALQUERY>
        <SPATIALFILTER relation="area_intersection">
          <ENVELOPE maxy="30" maxx="30" miny="0" minx="0" />
        </SPATIALFILTER>
      </SPATIALQUERY>
```

```
      <SIMPLERENDERER>
        <SIMPLEPOLYGONSYMBOL fillcolor="255,255,255" filltype="cross"
/>
      </SIMPLERENDERER>
    </LAYER>
  </GET_IMAGE>
</REQUEST>
</ARCXML>
```

SPATIALQUERY

Used in: **CONFIG** **REQUEST**
Servers: Image Query Feature Extract ArcMap
Parent elements: **BUFFER** **GET_FEATURES** **LAYER** **LAYERDEF**

<SPATIALQUERY

When using ArcMap Server:
featurelimit ="*integer*" **[All that meet criteria]**
searchorder ="optimize | spatialfirst | attributefirst" **[optimize]**
subfields ="#ALL# | #ID# | #SHAPE# | Other fields in database" **[#ALL#]**
where ="*string*"

When using Image, Extract, Query, or Feature Server:
accuracy ="Distance between points" **[0]**
featurelimit ="*integer*" **[All that meet criteria]**
joinexpression ="*string*"
jointables ="*string*"
searchorder ="optimize | spatialfirst | attributefirst" **[optimize]**
subfields ="#ALL# | #ID# | #SHAPE# | Other fields in database" **[#ALL#]**
where ="*string*"
>

When parent element is ***BUFFER:***
<SPATIALFILTER... />

When parent element is ***GET_FEATURES:***
<BUFFER... />
<FEATURECOORDSYS... />
<FILTERCOORDSYS... />
<SPATIALFILTER... />

When parent element is ***LAYER and LAYERDEF in GET_IMAGE:***
<BUFFER... />
<SPATIALFILTER... />

When parent element is ***LAYER in CONFIG:***
<FILTERCOORDSYS... />
<SPATIALFILTER... />

</SPATIALQUERY>

Description:

Queries a dataset in one of three ways:

1. A tabular query based on the value of attributes. For example, an attribute query asks for all cities in Canada where the population is greater than 500,000.
2. A spatial query based on features selected on a map. For example, a rectangle might be dragged over the eastern United States to select a group of cities.
3. A combination of a tabular and spatial query. For example, a group of cities might be selected, but only cities with a population greater than 500,000 are displayed.

SPATIALQUERY is also used to join DBF files to shapefiles and to relate tables in ArcSDE.

Restrictions:

- The *where* attribute is required when the *jointable* attribute for ArcSDE tables is used. Otherwise, this attribute is optional.
- The attributes *accuracy* and *featurelimit* are valid only in GET_FEATURES requests.
- A map configuration file with a SPATIALQUERY containing a full ArcSDE name for a field (TABLE.FIELD) works only in an ArcIMS service. The file cannot be read locally in ArcIMS Author, ArcExplorer 4, or the ArcIMS Java Viewers.
- When joining shapefiles, shapefile layers can only be joined to other DBF files located in the same directory as the shapefile. A joined DBF file cannot be another shapefile DBF file that is currently being used in an ArcIMS service.
- DBF jointable names are limited to 10 characters.
- When joining tables in ArcSDE, ArcSDE layers can only be joined to other tables in the relational database management system (RDBMS).

Notes:

- Without SPATIALFILTER, SPATIALQUERY works exactly the same as QUERY when querying database attributes.

- Queries defined in a map configuration file or generated in an ArcMap document always filter the data and cannot be changed through a request. Any requests made to a filtered layer in a map configuration file can select only features within the filtered subset. For example, assume a map configuration file has a world cities layer that has been filtered to display only cities with a population greater than 1,000,000. All requests to this layer, whether tabular or spatial, include only cities with a population greater than 1,000,000. Other cities in the database are ignored.

- **Using *subfields* in a map configuration file (Image and Feature Server).** The *subfields* attribute can be used to limit the number of available fields in a shapefile or ArcSDE layer along with any joined tables. In a SERVICEINFO response,

only the fields listed in *subfields* are included. In a FEATURES response, only the subfields listed are available for querying. The following rules apply:

- You should always include the #SHAPE# and #ID# fields in the list of subfields in the map configuration file. If #SHAPE# and #ID# are not included, a map will display, but queries will not work properly.
- If you plan to use CHARTSYMBOL, CHARTVALUE, SIMPLELABELRENDERER, VALUEMAPRENDERER, VALUEMAPLABELRENDERER, or TRUETYPEMARKERSYMBOL, you must include any fields used with these elements in the *subfields* list.
- If you are using ArcSDE layers, the fully qualified name must be used for all fields in the ArcSDE layer and any joined tables: ARCSDENAME.TABLE.NAME. Note that the map configuration file cannot be read locally in ArcIMS Author or ArcExplorer 4 when using fully qualified names.

- In a GET_FEATURES request, one of the options is to return the geometry of selected features by setting the *geometry* attribute to true. In addition, the #SHAPE# field (or #ALL#) must be included in the SPATIALFILTER *subfields* list.

```
<GET_FEATURES outputmode="XML" geometry="true">
<LAYER id="4" />
<SPATIALQUERY subfields="#ALL#" where="NAME='Los Angeles'" />
</GET_FEATURES>
```

- When used in GET_FEATURES, queries and finds can be made case insensitive by including the UPPER parameter around the field name. Note that the value must have all uppercase letters.

  An example for a shapefile:

```
<SPATIALQUERY subfields="#ALL#" where="UPPER(NAME)="TORONTO" >
```

  An example for an ArcSDE layer:

```
<SPATIALQUERY subfields="#ALL#"
where="UPPER(MYSDE.US_STATES.STATE_NAME)="FLORIDA" >
```

- When creating a *where* clause, two fields from the same attribute table can be used for comparison. For example, a query can be made to find states with a female population greater than the male population.

```
<SPATIALQUERY where="COUNTRY.STATE.FEMALES &gt;
COUNTRY.STATE.MALES" />
```

- Some symbols must be "escaped" inside a where expression:
  ampersand (&) is escaped to &
  double quotes (") are escaped to "
  single quotes (') are escaped to '

greater than (>) is escaped to >
less than (<) is escaped to <

- The following operators work in a where clause: =, >, >=, <, <=, <>, LIKE, BETWEEN, IN, NOT IN. The following are not valid: ORDER BY and DISTINCT.

- By default query and find operations are case sensitive. In many cases, a case *in*sensitive query is needed because what a user types in is unpredictable. Most data sources support UPPER within a query. For a case insensitive query, use UPPER around the field name and convert the comparison text to all upper case letters.

```
<SPATIALQUERY subfields="#ALL#" where="UPPER(NAME) = 'CANADA'">
```

  UPPER does not work with all data sources, including Personal Geodatabases and ArcSDE for Coverages.

- **Joining ArcSDE tables.** When joining ArcSDE tables, all valid joins between the RDBMS and ArcSDE are also valid in ArcIMS. To join ArcSDE tables, a where clause is used for defining which tables are joined and for any additional filters using SQL syntax. Multiple joins are also permitted.

  In the following example, the where statement includes setting up two relates (in bold type) and filtering the selection to FIPS_CNTRY='CA' (in italic type).

```
<SPATIALQUERY where="DB.CITY.CITY_FIPS = DB.SCHOOLS.CITY_FIPS and
DB.SCHOOLS.SCHOOL_ID = DB.SCHOOL_STATS.SCHOOL_ID and
DB.CITY.CNTY_FIPS=&apos;013&apos;" jointables="DB.SCHOOLS
DB.SCHOOL_INFO" />
```

  In this statement, cities (DB.CITY) are first joined to schools (DB.SCHOOLS) using the column CITY_FIPS as the common join item. Next, the schools (DB.SCHOOLS) are joined to a table with statistics on the schools (DB.SCHOOL_STATS) using SCHOOL_ID as the join item. A further filter is placed on this query by limiting the results to schools located within a specified county, in this example, where CNTY_FIPS='013'.

  The list of joined tables must be listed under the *jointables* attribute. If more than one table is joined, table names are separated by a space. When naming tables, the full ArcSDE table name must be used. In the above example, the two joined tables are DB.SCHOOLS and DB.SCHOOL_INFO.

- **Joining DBF files.** When joining DBF tables, joins can be made between the shapefile DBF table and one or more external DBF tables that reside in the same directory as the shapefile. One restriction is that a joined DBF file cannot be another shapefile DBF file that is currently being used in an ArcIMS service.

  To join DBF files, the *joinexpression* attribute is used. This attribute uses the

syntax: "To=[master table column which will be used for joining], From=[defines a join table column which will be joined], Type=[exact or scan]".

To	Refers to the master DBF table and defines the field that is used for joining. When referring to this table, the DBF table name must be used as a prefix to the field name. The entire expression is surrounded by square brackets, e.g., joinexpression="To=[mastertable.fieldname]".
From	Refers to the DBF table that is joined to the master DBF table and the field that is used for joining, e.g., joinexpression="From=[jointable.fieldname]".
Type=[exact]	Defines an exact match relation that permits only a single match between the master and join tables. Both one-to-one and many-to-one relations are exact match relations. In a one-to-one relation, there is only one record in the master that matches a single record in the join table. In a many-to-one relation, there are one or more records in the master that match a single join record. If there are multiple join records that match a single master record, then a composite record is only generated for the first join record.
Type=[scan]	In a scan relation, if there are multiple join records for a master record, there is one composite record in the extended data file for each of the matching join records. Both one-to-many and many-to-many relations are scan relations. In a one-to-many relation, each record in the master can have multiple matching join records. A many-to-many relation is the same as one-to-many, except that different master records can match the same join record.

To, From, and Type parameters are case sensitive (first letter is capitalized), and they are separated by a comma, ",". Any number of joined tables can be defined in the *joinexpression* attribute. Joined tables are separated by a semicolon, ";" such as the following:

```
joinexpression="To=[A.ID],From=[B.ID],Type=[scan];To=[B.NAME],From=[C.NAME],
Type=[exact] "
```

All joined DBF tables must be listed under *jointables*. If multiple tables are joined, table names are separated by a space. When naming tables, the name of the DBF file without the extension is used. An example DBF query expression is:

```
<SPATIALQUERY
joinexpression="To=[counties.CNTY_FIPS],From=[countyinfo.FIPS],Type=[scan];
To=[countyinfo.FIPS],From=[state_roads.FIPS],Type=[scan]"
jointables="countyinfo state_roads" where="counties.
NAME=&amp;apos;Washoe&amp;apos;" />
```

In this example, a county DBF file (counties.dbf) is first joined to a DBF file containing county information (countyinfo.dbf). They are joined on the field FIPS (counties.CNTY_FIPS and countyinfo.FIPS). Next, countyinfo.dbf is joined to state_roads.dbf, once again using FIPS as the join item. The jointables are listed as countyinfo and state_roads, separated by a space. The query is filtered to only include one county, in this case Washoe.

When joining DBF files to a shapefile, the DBF file cannot be read-only. An indication of this problem is the FIELD information from the joined tables shows up in SERVICEINFO, but the data is not drawn on the map.

- **Querying with dates**. The syntax for querying dates is the same regardless of the locale. A date query uses the following syntax:

  {ts 'YYYY-MM-DD hh:mi:ss'}

  where

YYYY	Year	Required	Use four digits for the year.
MM	Month (01-12)	Required	Use two digits for the month. March is 03.
DD	Day (01-31)	Required	Use two digits for the day. The fourth is 04.
hh	Hour (00-23)	Optional	Use a 24-hour clock. 8 a.m. is 08, and 8 p.m. is 20.
mi	Minutes (00-59)	Optional	Use two digits for the minutes. If minutes is used, hours must also be included.
ss	Seconds (00-59)	Optional	Use two digits for the seconds. If seconds is used, hours and minutes must also be included.

  The year, month, and day are each separated by a dash (-). The hour, minutes, and seconds are each separated by a colon (:). The date is enclosed in single quotes (') inside curly brackets ({}). Before the date, ts (for time stamp) must be included.

  For 8:03:23 a.m. January 4, 2000, the query on a DBF file looks like:

```
<SPATIALQUERY where="MYDATE = {ts '2000-01-04 08:03:32'}" />
```

  For 9:18 p.m. March 8, 2002, the query on an ArcSDE layer looks like:

```
<SPATIALQUERY where="ARCSDE.TABLE.MYDATE = {ts '2002-03-08
21:18:00'}" />
```

Attribute Descriptions for SPATIALQUERY:

Attribute	Usage
accuracy	Used in GET_FEATURES requests. Within a feature, generalizes points based on the distance specified and the resolution of the image. Units are the same as the service.
featurelimit	Used in GET_FEATURES requests. Maximum number of features to be extracted that meet criteria. The value for this featurelimit overrides any featurelimit set in GET_FEATURES.
joinexpression	Used for join tables with DBF files only; *jointables* must be filled to contain list of tables used; not required when a jointable is done on ArcSDE. String must form expression: "To=[master table column which will be used for joining], From=[defines a join table column which will be joined], Type=[exact or scan]".
jointables	List of joined table names separated by blank spaces; for ArcSDE, table name is full name including database name (e.g., DATA.STATE); for shapefiles, use the DBF filename without the extension (e.g., STATES).
searchorder	Used with ArcSDE layers only. Determines whether the attribute or spatial part of an ArcSDE query is processed first. "Spatialfirst" processes the spatial part of the query before the attribute part. "Attributefirst" processes the attribute part of the query first. If "optimize" is used, ArcSDE will make the judgment whether to pick "spatialfirst" or "attributefirst".
subfields	List of fields available for querying or extracting. Multiple fields can be included in the *subfields* list. Fields must be separated by blank space. If *subfields* is not used, all fields are returned. If *subfields* is used, only listed fields are returned. The subfields #SHAPE# or #ALL# must be included if geometry is to be returned in a FEATURES response. In addition, the GET_FEATURES *geometry* attribute must be set to true. The *subfields* list can include fields from the layer table or a joined table. • For shapefiles with no joined tables, the field can be referenced using the short format. field="AREA" • For shapefiles with joined tables, the name of the joined table must be included along with the field. field="JOINEDTABLE.AREA" • For ArcSDE layers with or without joined tables, the field must be referenced using the full long format.

	field="ARCSDENAME.TABLE.AREA"
where	Defines "where" part of SQL expression. Required when *jointables* attribute for ArcSDE tables is used. See Notes section for information on querying dates.

Examples for SPATIALQUERY:

Example 1: Setting a spatial query in a GET_IMAGE request. Note in this example that the DATASET fromlayer is "Countries". This refers to the LAYER id in the map configuration file, not the LAYER name.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_IMAGE>
    <PROPERTIES>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      <IMAGESIZE width="643" height="502" />
    </PROPERTIES>
    <LAYER type="featureclass" name="select layer" visible="true"
id="300">
      <DATASET fromlayer="Countries" />
      <SPATIALQUERY>
        <SPATIALFILTER relation="area_intersection">
          <ENVELOPE maxy="30" maxx="30" miny="0" minx="0" />
        </SPATIALFILTER>
      </SPATIALQUERY>
      <SIMPLERENDERER>
        <SIMPLEPOLYGONSYMBOL fillcolor="255,255,255" filltype="cross"
/>
      </SIMPLERENDERER>
    </LAYER>
  </GET_IMAGE>
</REQUEST>
</ARCXML>
```

Example 2: Setting a spatial query in a GET_FEATURES request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
    <REQUEST>
      <GET_FEATURES outputmode="xml" geometry="false">
        <LAYER id="Countries" />
           <SPATIALQUERY >
             <SPATIALFILTER relation="area_intersection">
                <POLYGON>
                <RING>
                   <POINT x="-87.73640582356195" y="41.84726275" />
                   <POINT x="-87.73640582356195" y="41.884308250000004"
/>
```

```
                    <POINT x="-87.68764017643805" y="41.884308250000004"
/>
                    <POINT x="-87.68764017643805" y="41.84726275" />
                    <POINT x="-87.73640582356195" y="41.84726275" />
                  </RING>
                  </POLYGON>
              </SPATIALFILTER>
            </SPATIALQUERY>
      </GET_FEATURES>
    </REQUEST>
</ARCXML>
```

Example 3: Joining DBF files in CONFIG.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.215027" miny="18.924782" maxx="-66.969849"
maxy="71.406647" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>" />
      </WORKSPACES>

      <LAYER type="featureclass" name="counties" visible="true" id="0">
        <DATASET name="COUNTIES" type="polygon" workspace="shp_ws-0" />

          <SPATIALQUERY
where="counties.STATE_NAME=&amp;apos;Nevada&amp;apos;" jointables="countyinfo"
joinexpression="To=[counties.FIPS],From=[countyinfo.FIPS],Type=[scan]"
>
            <SPATIALFILTER relation="area_intersection">
                  <ENVELOPE minx="-126" miny="31" maxx="-108" maxy="46"
/>
            </SPATIALFILTER>
          </SPATIALQUERY>

        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="27,127,27" filltype="solid"
/>
        </SIMPLERENDERER>
      </LAYER>

    </MAP>
  </CONFIG>
</ARCXML>
```

Example 4: Joining ArcSDE tables in CONFIG.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-175.2" miny="-90.0" maxx="179.2" maxy="83.6"
name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
      <SDEWORKSPACE name="sde_ws-4" server="sierra" instance="esri_sde"
database="" user="world_data" encrypted="true" password="LXEMUR" />
      </WORKSPACES>
      <LAYER type="featureclass" name="WORLD.CITY" visible="true"
id="0">
      <DATASET name="WORLD.CITY" type="point" workspace="sde_ws-4" />

        <SPATIALQUERY where="WORLD.CITY.FIPS_CNTRY =
WORLD.COUNTRYP.FIPS_CNTRY and WORLD.COUNTRYP.FIPS_CNTRY=&amp;apos;CA&amp;apos;
" jointables="WORLD.COUNTRYP" >
          <SPATIALFILTER relation="area_intersection">
           <ENVELOPE maxy="30" maxx="30" miny="0" minx="0" />
          </SPATIALFILTER>
        </SPATIALQUERY>

      <SIMPLERENDERER>
        <SIMPLEMARKERSYMBOL color="27,227,27" width="8" />
      </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 5: Using both an attribute query and a spatial filter in CONFIG.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP dynamic="true">
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
```

```
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CITIES" visible="true" id="2">
      <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
      <SPATIALQUERY where="POPULATION &gt; 2000000">
        <SPATIALFILTER relation="area_intersection">
          <ENVELOPE maxy="30" maxx="30" miny="0" minx="0" />
        </SPATIALFILTER>
      </SPATIALQUERY>
      <SIMPLERENDERER>
        <SIMPLEMARKERSYMBOL type="square" width="5" />
      </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 6: Using a date query in a GET_FEATURES request.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES featurelimit="25" beginrecord="0" outputmode="xml"
geometry="false" envelope="true" compact="true">
      <LAYER id="0" />
      <SPATIALQUERY subfields="#ALL#" where="MYDATE = {ts '2000-01-
07'}" />
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

SQVAR

Used in: CONFIG RESPONSE
Servers:
Parent elements: STOREDQUERY

<SQVAR

name ="[%var%]"
position ="nonnegative value"
>

<**FIELD...** />

</SQVAR >

Bold: Attribute or child element is required.

Description:

Defines a position of the query field in the *where* expression of the query element.

Restrictions:

- Used only with the ArcIMS Java Viewers and ArcExplorer 4.

Notes:

None

Attribute Descriptions for SQVAR:

Attribute	Usage
name	Defines string used for the variable name.
position	Defines the location in the string of the opening bracket "[" in the variable expression. The value should be equal to or less than the position in the string where the bracket is located. If "'" is used in the expression, this is counted as one character.

Examples for SQVAR:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
```

```
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-141.0030059814453" miny="41.9133186340332"
maxx="-52.62028121948242" maxy="83.10832214355469"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to CANADA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="province" visible="true" id="0">
        <DATASET name="province" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltransparency="1.0"
fillcolor="127,127,127" />
        </SIMPLERENDERER>
        <EXTENSION type="StoredQuery">
          <STOREDQUERIES>
            <STOREDQUERY name="Provinces">
              <QUERY where="( NAME = &apos;[%var%]&apos; )"
subfields="#SHAPE# AREA CODE NAME POP1991 POP91_SQMI" />
              <SQVAR position="0" name="[%var%]">
                <FIELD name="NAME" precision="0" type="12" size="25" />
              </SQVAR>
            </STOREDQUERY>
          </STOREDQUERIES>
      </EXTENSION>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

STOREDQUERIES

Used in: CONFIG RESPONSE
Servers:
Parent elements: EXTENSION

<STOREDQUERIES >
No Attributes

(m) <STOREDQUERY... />

</STOREDQUERIES >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

The main element used around one or more STOREDQUERY expressions in a stored query EXTENSION.

Restrictions:

- Used only with the ArcIMS Java Viewers and ArcExplorer 4.

Notes:

None

Examples for STOREDQUERIES:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-141.0030059814453" miny="41.9133186340332"
maxx="-52.62028121948242" maxy="83.10832214355469"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to CANADA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="province" visible="true" id="0">
        <DATASET name="province" type="polygon" workspace="shp_ws-0" />
```

```
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltransparency="1.0"
fillcolor="127,127,127" />
        </SIMPLERENDERER>
        <EXTENSION type="StoredQuery">
          <STOREDQUERIES>
            <STOREDQUERY name="Provinces">
              <QUERY where="( NAME = &apos;[%var%]&apos; )"
subfields="#SHAPE# AREA CODE NAME POP1991 POP91_SQMI" />
              <SQVAR position="0" name="[%var%]">
                <FIELD name="NAME" precision="0" type="12" size="25" />
              </SQVAR>
            </STOREDQUERY>
          </STOREDQUERIES>
      </EXTENSION>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

STOREDQUERY

Used in: CONFIG RESPONSE
Servers:
Parent elements: STOREDQUERIES

<STOREDQUERY

name ="*string*"
>

<**QUERY... /**>
<SQVAR... />

</STOREDQUERY>

Bold: Attribute or child element is required.

Description:

Defines data for a particular stored query.

Restrictions:

- Used only with the ArcIMS Java Viewers and ArcExplorer 4.

Notes:

None

Attribute Descriptions for STOREDQUERY:

Attribute	Usage
name	Stored query name.

Examples for STOREDQUERY:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-141.0030059814453" miny="41.9133186340332"
maxx="-52.62028121948242" maxy="83.10832214355469"
```

```
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to CANADA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="province" visible="true" id="0">
        <DATASET name="province" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltransparency="1.0"
fillcolor="127,127,127" />
        </SIMPLERENDERER>
        <EXTENSION type="StoredQuery">
          <STOREDQUERIES>
            <STOREDQUERY name="Provinces">
              <QUERY where="( NAME = &apos;[%var%]&apos; )"
subfields="#SHAPE# AREA CODE NAME POP1991 POP91_SQMI" />
              <SQVAR position="0" name="[%var%]">
                <FIELD name="NAME" precision="0" type="12" size="25" />
              </SQVAR>
            </STOREDQUERY>
          </STOREDQUERIES>
      </EXTENSION>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

SUBSET

Used in: REQUEST
Servers: Metadata (Browse)
Parent elements: SEARCH_METADATA

```
<SUBSET

    docid ="string"
    type  ="children | parents | siblings | ancestors | descendants"
>

    No Child Elements
</SUBSET >
```

Bold: Attribute or child element is required.

Description:

Specifies a subset of the documents in the metadata repository to be searched.

Restrictions:

None

Notes:

- By default, all documents in a metadata repository are searched. When SUBSET is included, *name*, *owner*, and *userid* attributes specify the "root document" of the search, and the *type* attribute specifies the relation of the desired documents to the root document.

Attribute Descriptions for SUBSET:

Attribute	Usage
docid	String used to uniquely identify a document. The client used to publish the metadata is responsible for creating the document ID. This ID is automatically assigned when using ArcCatalog. If another client is used, GET_UUID can be used to request a valid ID. The format for an ID is the following: {HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH} where H is a hexadecimal digit (0-9,a-f,A-F). The ID is limited to 38 characters.
type	Target document's relationship to the source document.

Examples for SUBSET:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <SEARCH_METADATA>
        <SUBSET type="children" docid="{F7DDF21-BC01-4C20-8AA5-
243B33ED0B1E}" />
     </SEARCH_METADATA>
   </GET_METADATA>
  </REQUEST>
</ARCXML>
```

TABLE_NAME

Used in: CONFIG
Parent elements: METADATA_CONFIG

```
<TABLE_NAME

    prefix ="string"  [imsmetadata]
>

    No Child Elements
</TABLE_NAME>
```

Description:

Sets the prefix for metadata tables.

Restrictions:

None

Notes:

- This element is not available from the ArcIMS Author interface.

Attribute Descriptions for TABLE_NAME:

Attribute	Usage
prefix	Prefix name for metadata tables. Eight RDBMS tables are created to store metadata. The prefix uniquely identifies these metadata tables. The default prefix is "imsmetadata". Using this prefix, the main internal metadata table is then named "imsmetadat". The other tables would be named imsmetadatarelatoinships, imsmetadataelements, imsmetadatathumbnails, imsmetadatausers, imsmetadatavalues, imsmetadatawordindex, and imsmetadatawords.

Examples for TABLE_NAME:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
    </ENVIRONMENT>
    <METADATA_CONFIG>
      <WORKSPACES>
```

```
        <SDEWORKSPACE name="unique_name" server="server_name"
instance="port:5151" database="optional_database_name" user="user_name"
password="user_password" />
      </WORKSPACES>
      <METADATA_CONTENT validate="true" />
      <TABLE_NAME prefix="imsmetadata" />
    </METADATA_CONFIG>
  </CONFIG>
</ARCXML>
```

TAG

Used in: RESPONSE
Servers: Metadata (Browse)
Parent elements: COLLECTION_INFO

<TAG

value ***="string"***
>

No Child Elements
</TAG >

Bold: Attribute or child element is required.

Description:

Identifies an element in an XML document.

Restrictions:

None

Notes:

None

Attribute Descriptions for TAG:

Attribute	Usage
value	Defines the location of an element inside an XML tree. For example, the element "title" might be inside an XML structure such as <idinfo> <citation> <citeinfo> **<title>...</title>** </citeinfo> </citation> </idinfo> To define "title", the string value for this attribute would be "idinfo/citation/citeinfo/**title**".

Examples for TAG:

Example 1:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <METADATA>
      <COLLECTION_INFO>
        <TAG value="metadata/Binary/Thumbnail/Data" />
        <TAG value="metadata/dataqual/attracc/attraccr" />
        <TAG value="metadata/dataqual/complete" />
         ...
      </COLLECTION_INFO>
    </METADATA>
  </RESPONSE>
</ARCXML>
```

TAGTEXT

Used in: REQUEST
Servers: Metadata (Browse)
Parent elements: SEARCH_METADATA

<TAGTEXT

tag ="***string***"
word ="***string***"
zcode ="***string***"
attribute ="*string*"
>

No Child Elements
</TAGTEXT >

Bold: Attribute or child element is required.

Description:

Searches text within a specified element in metadata documents.

Restrictions:

- Either *tag* or *zcode* must be specified, but not both.
- *Attribute* is valid only with *tag*. It is ignored when *zcode* is used.

Notes:

- A TAGTEXT search can be further restricted by limiting the search to an individual attribute.
- For searching numeric values, see TAGVALUE.

Attribute Descriptions for TAGTEXT:

Attribute	Usage
attribute	Name of an attribute for searching.
tag	Defines the location of an element inside an XML tree. For example, the element "title" might be inside an XML structure such as <idinfo> <citation> <citeinfo> **<title>...</title>** </citeinfo> </citation> </idinfo>

	To define "title", the string value for this attribute would be "idinfo/citation/citeinfo/**title**".
word	List of one or more keywords. Words are separated by a space.
zcode	Numeric code defining semantics from the GEO Attribute Set.

Examples for TAGTEXT:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <SEARCH_METADATA operator="and">
        <ENVELOPE minx="-176.98" miny="18.93" maxx="-66.97"
maxy="71.35" spatialoperator="within" />
        <TAGVALUE
tag="metadata/dataqual/lineage/srcinfo/srccite/citeinfo/pubdate"
greaterthan="1990" />
        <FULLTEXT word="cities"/>
        <SEARCH_METADATA operator="or">
          <TAGTEXT tag="metadata/idinfo/citation/citeinfo/geoform"
word="data"/>
          <TAGTEXT tag="metadata/idinfo/citation/citeinfo/geoform"
word="digital"/>
        </SEARCH_METADATA>
      </SEARCH_METADATA>
    </GET_METADATA>
  </REQUEST>
</ARCXML>
```

TAGVALUE

Used in: REQUEST
Servers: Metadata (Browse)
Parent elements: SEARCH_METADATA

```
<TAGVALUE

    equalto ="string"
    greaterthan ="string"
    greaterthanorequalto ="string"
    lessthan ="string"
    lessthanorequalto ="string"
    tag ="string"
    zcode ="string"
    attribute ="string"
>

    No Child Elements
</TAGVALUE >
```

Bold: Attribute or child element is required.

Description:

Searches numeric values within a specified element in a metadata document.

Restrictions:

- Either *tag* or *zcode* must be specified, but not both.
- *Attribute* is valid only with *tag*. It is ignored when *zcode* is used.
- One or two of the operator attributes from the following list must be specified: *lessthan*, *greaterthan*, *lessthanorequalto*, *greaterthanorequalto*, or *equalto*. If two attributes are specified, they are automatically concatenated together with an "and" operator.

Notes:

- For searching text values, see TAGTEXT or FULLTEXT.

Attribute Descriptions for TAGVALUE:

Attribute	Usage
attribute	Name of an attribute for searching.
equalto	Equal to operator for comparing two values.
greaterthan	Greater than operator for comparing two values.
greaterthanorequalto	Greater than or equal to operator for comparing two values.
lessthan	Less than operator for comparing two values.

lessthanorequalto	Less than or equal to operator for comparing two values.
tag	Defines the location of an element inside an XML tree. For example, the element "title" might be inside an XML structure such as <idinfo> <citation> <citeinfo> **<title>...</title>** </citeinfo> </citation> </idinfo> To define "title", the string value for this attribute would be "idinfo/citation/citeinfo/**title**".
zcode	Numeric code defining semantics from the GEO Attribute Set.

Examples for TAGVALUE:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <SEARCH_METADATA operator="and" maxresults="10" startresult="0" >
        <TAGVALUE tag="metadata/idinfo/citation/citeinfo/pubdate"
greaterthanorequalto="20001006" lessthanorequalto="20010104"/>
      </SEARCH_METADATA>
    </GET_METADATA>
  </REQUEST>
</ARCXML>
```

TARGETLAYER

Used in: **REQUEST**
Servers: Image Query Feature ArcMap
Parent elements: **BUFFER**

<TARGETLAYER

id ="***string***"
>

No Child Elements
</TARGETLAYER >

Bold: Attribute or child element is required.

Description:

Defines a target layer for selecting features based on a buffer in the same or different layer in the ArcIMS service.

Restrictions:

- Must refer to an existing layer ID in the ArcIMS service.

Notes:

None

Attribute Descriptions for TARGETLAYER:

Attribute	Usage
id	Reference to unique layer ID as defined in the ArcIMS service.

Examples for TARGETLAYER:

Example 1: When in a GET_FEATURES request.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_FEATURES featurelimit="25" beginrecord="0" outputmode="xml"
geometry="FALSE" envelope="true" compact="true">
      <LAYER id="2" /> <!-- states -->
      <SPATIALQUERY subfields="#SHAPE# NAME">
        <BUFFER distance="1" >
          <TARGETLAYER id="4" /> <!-- cities -->
          <SPATIALQUERY subfields="NAME" />
        </BUFFER>
        <SPATIALFILTER relation="area_intersection" >
          <ENVELOPE minx="-16154208.3772906" miny="-4165319.9729724"
maxx="-4904885.23874079" maxy="4271672.38093997" />
```

```
        </SPATIALFILTER>
      </SPATIALQUERY>
    </GET_FEATURES>
  </REQUEST>
</ARCXML>
```

Example 2: When in a GET_IMAGE request. Note in this example that the DATASET fromlayer is "CITIES". This refers to the LAYER id in the map configuration file, not the LAYER name.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="867805.08306143" miny="6771243.45626185"
maxx="1576642.8482793" maxy="7302871.78017525"/>
      </PROPERTIES>
      <LAYER type="featureclass" name="theBufferTarget" visible="true"
id="buffertarget">
        <DATASET fromlayer="CITIES"/>
        <SPATIALQUERY>
          <BUFFER distance="100" bufferunits="miles">
            <TARGETLAYER id="CITIES"/>
          </BUFFER>
          <SPATIALFILTER relation="area_intersection">
            <ENVELOPE minx="867805.08306143" miny="6771243.45626185"
maxx="1576642.8482793" maxy="7302871.78017525"/>
          </SPATIALFILTER>
        </SPATIALQUERY>
        <SIMPLERENDERER>
          <SIMPLEMARKERSYMBOL color="255,0,0" />
        </SIMPLERENDERER>
      </LAYER>
      <LAYER type="featureclass" name="theBuffer" visible="true"
id="buffer">
        <DATASET fromlayer="CITIES"/>
        <SPATIALQUERY>
          <BUFFER distance="100" bufferunits="miles" />
          <SPATIALFILTER relation="area_intersection">
            <ENVELOPE minx="867805.08306143" miny="6771243.45626185"
maxx="1576642.8482793" maxy="7302871.78017525"/>
          </SPATIALFILTER>
        </SPATIALQUERY>
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="100,100,100" filltype="solid"
filltransparency="0.5" />
        </SIMPLERENDERER>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

TEXT

Used in: **CONFIG** **REQUEST**
Servers: Image ArcMap
Parent elements: **OBJECT**

```
<TEXT

    coords ="double"
    label ="string"
>

    <TEXTMARKERSYMBOL... />

</TEXT >
```

Bold: Attribute or child element is required.

Description:

Places text on an acetate layer.

Restrictions:

- TEXT, as an object in an acetate layer, can only be used with Image Services in an HTML viewer.

Notes:

- TEXT is used only in acetate layers. To label features, use TEXTSYMBOL.

Attribute Descriptions for TEXT:

Attribute	Usage
coords	Text placement location. Coordinate pair is separated by white space by default. The separator can be changed by using SEPARATORS. If using pixel coordinates, "0 0" is in the lower left corner of the map viewer area.
label	Text label.

Examples for TEXT:

Example 1: When in an acetate layer of a map configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
      <ENVELOPE minx="-141.003006" miny="41.913319" maxx="-52.620281"
maxy="83.108322" name="Initial_Extent" />
      <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-16" directory="<path to CANADA
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="province" visible="true" id="0">
      <DATASET name="province" type="polygon" workspace="shp_ws-16" />
      <SIMPLERENDERER>
        <SIMPLEPOLYGONSYMBOL fillcolor="227,127,227" filltype="solid"
/>
      </SIMPLERENDERER>
      </LAYER>
      <LAYER type="acetate" name="Selectedmark" id="acetate">
      <OBJECT units="pixel">
          <TEXT coords="100 100" label="Using text in an Acetate
layer">
             <TEXTMARKERSYMBOL fontstyle="regular" fontsize="30"
font="Times New Roman" />
          </TEXT>
      </OBJECT>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When in an acetate layer of a GET_IMAGE request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
  <GET_IMAGE>
    <PROPERTIES>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
    </PROPERTIES>
    <LAYER type="acetate" name="acetate" id="acetate">
      <OBJECT units="pixel">
      <TEXT coords="100 100" label="You are here">
         <TEXTMARKERSYMBOL font="Arial" />
```

```
      </TEXT>
      </OBJECT>
    </LAYER>
  </GET_IMAGE>
</REQUEST>
</ARCXML>
```

TEXTMARKERSYMBOL

Used in: **CONFIG** **REQUEST**
Servers: Image ArcMap
Parent elements: **TEXT**

<TEXTMARKERSYMBOL

When using ArcMap Server:
blockout ="0,0,0 - 255,255,255"
font ="Any system font" **[Arial]**
fontcolor ="0,0,0 - 255,255,255" **[0,0,0]**
fontsize ="1 - NNN" **[10]**
fontstyle ="regular | bold | italic | underline | bolditalic" **[regular]**
halignment ="left | center | right" **[right]**
outline ="0,0,0 - 255,255,255"
valignment ="top | center | bottom" **[top]**

When using Image or Feature Server:
angle ="0.0 - 360.0" **[0]**
antialiasing ="true | false" **[false]**
blockout ="0,0,0 - 255,255,255"
font ="Any system font" **[Arial]**
fontcolor ="0,0,0 - 255,255,255" **[0,0,0]**
fontsize ="1 - NNN" **[10]**
fontstyle ="regular | bold | italic | underline | outline | bolditalic" **[regular]**
glowing ="0,0,0 - 255,255,255"
halignment ="left | center | right" **[right]**
interval ="0 - NNN" **[0]**
outline ="0,0,0 - 255,255,255"
overlap ="true | false" **[true]**
printmode ="titlecaps | allupper | alllower | none" **[none]**
transparency ="0.0 - 1.0" **[1.0]**
valignment ="top | center | bottom" **[top]**
>

No Child Elements
</TEXTMARKERSYMBOL >

Description:

Adds static text to an acetate layer.

Restrictions:

- TEXTMARKERSYMBOL is part of an object in an acetate layer and can only be used with Image Services in an HTML viewer.
- *Outline* and *glowing* should not be used together; use one or the other.

Notes:

- TEXTMARKERSYMBOL is used only in acetate layers. To label features, use TEXTSYMBOL.
- This element is not available from the ArcIMS Author interface.

Attribute Descriptions for TEXTMARKERSYMBOL:

Attribute	Usage
angle	Angle of rotation in degrees moving counterclockwise; 0 degrees is horizontal.
antialiasing	Used to make edges of labels and symbols smoother. When set to "true", antialiasing is active. Note that the time to generate a map may significantly increase when antialiasing is turned on.
blockout	Provides a background behind text. Select color using RGB values.
font	Font name. The name is case sensitive. If font name uses "&", use "&" instead. For example, ESRI Transportation & Civic should be written as ESRI Transportation & Civic. For Feature Services, the font must reside on the client machine or else the system default font is used.
fontcolor	Font color using RGB values.
fontsize	Font size.
fontstyle	Font style.
glowing	Glow color around text using RGB values.
halignment	Horizontal alignment of label compared to label point.
interval	Distance between point and printed label.
outline	Outline color using RGB values.
overlap	Determines if labels can overlap this symbol. When "true", labels can overlap. When "false", labels will not overlap the symbol. If labels are not drawing as expected, check if *overlap* is set to "false" for this symbol or any other symbol in the ArcIMS service.
printmode	Determines how labels are printed. If "none" is used, no change is made to the label: Welcome to ArcIMS. If "alllower" is used, all letters are lowercase: welcome to arcims. If "allupper" is used, all letters are uppercase: WELCOME TO ARCIMS. If "titlecaps" is used, the first letter of each word in a label is uppercase and everything else is lowercase: Welcome To Arcims.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.

valignment Vertical alignment of label compared to label point.

Examples for TEXTMARKERSYMBOL:

Example 1: When in an acetate layer in a map configuration file or REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
      <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      <IMAGESIZE width="643" height="502" />
      </PROPERTIES>
      <LAYER type="acetate" name="acetate" id="acetate">
      <OBJECT units="pixel">
          <TEXT coords="100 100" label="You are here">
            <TEXTMARKERSYMBOL font="Arial" />
        </TEXT>
      </OBJECT>
      </LAYER>
    </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

TEXTSYMBOL

Used in: CONFIG REQUEST RESPONSE
Servers: Image Feature
Parent elements: EXACT OTHER RANGE SIMPLELABELRENDERER

```
<TEXTSYMBOL

    antialiasing ="true | false"  [false]
    blockout ="0,0,0 - 255,255,255"
    font ="Any system font"  [Arial]
    fontcolor ="0,0,0 - 255,255,255"  [0,0,0]
    fontsize ="1 - NNN"  [12]
    fontstyle ="regular | bold | italic | underline | outline | bolditalic"  [regular]
    glowing ="0,0,0 - 255,255,255"
    interval ="0 - NNN"  [0]
    outline ="0,0,0 - 255,255,255"
    printmode ="titlecaps | allupper | alllower | none"  [none]
    shadow ="0,0,0 - 255,255,255"
    transparency ="0.0 - 1.0"  [1.0]
>

    No Child Elements
</TEXTSYMBOL >
```

Description:

Symbol used to label point, line, and polygon layers.

Restrictions:

- *Outline* and *glowing* should not be used together; use one or the other.
- Not valid with ArcMap Server.

Notes:

- TEXTSYMBOL is used to label features in layers. To add text to an acetate layer, use TEXT.

Attribute Descriptions for TEXTSYMBOL:

Attribute	Usage
antialiasing	Used to make edges of labels and symbols smoother. When set to "true", antialiasing is active. Note that the time to generate a map may significantly increase when antialiasing is turned on.
blockout	Provides a background behind text. Select color using RGB values.
font	Font name. The name is case sensitive. If font name uses "&", use "&" instead. For example, ESRI Transportation & Civic should be

	written as ESRI Transportation & Civic. For Feature Services, the font must reside on the client machine or else the system default font is used.
fontcolor	Font color using RGB values.
fontsize	Font size.
fontstyle	Font style.
glowing	Glow color around text using RGB values.
interval	Distance in pixels from point 0.
outline	Outline color using RGB values.
printmode	Determines how labels are printed. If "none" is used, no change is made to the label: Welcome to ArcIMS. If "alllower" is used, all letters are lowercase: welcome to arcims. If "allupper" is used, all letters are uppercase: WELCOME TO ARCIMS. If "titlecaps" is used, the first letter of each word in a label is uppercase and everything else is lowercase: Welcome To Arcims.
shadow	Shadow color using RGB values.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.

Examples for TEXTSYMBOL:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="0">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <SIMPLELABELRENDERER field="NAME">
            <TEXTSYMBOL transparency="0.8" printmode="titlecaps"
antialiasing="true" font="Courier New" fontstyle="bolditalic"
fontsize="12" glowing="255,0,255" shadow="255,200,0"
fontcolor="0,255,100" blockout="124,124,124" interval="3" />
```

```
          </SIMPLELABELRENDERER>
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL filltransparency="1.0"
filltype="solid" fillcolor="227,27,27" boundarytype="solid"
boundarywidth="1" boundarycolor="0,0,0" />
          </SIMPLERENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

THUMBNAIL

Used in: REQUEST
Servers: Metadata (Publish)
Parent elements: PUT_METADATA

<THUMBNAIL >
No Attributes

No Child Elements
</THUMBNAIL >

Description:

Contains Base-64 encoding of a thumbnail image.

Restrictions:

None

Notes:

None

Examples for THUMBNAIL:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <PUBLISH_METADATA>
      <PUT_METADATA name="World" userid="{C64D8F38-82B4-11D5-99C2-
000086460FA0}" private="false" folder="true" parentname="Home"
parentowner="admin1" parentuserid="root" >
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
        <THUMBNAIL>...</THUMBNAIL>
        <!--<?xml version="1.0"?>
            <metadata>...</metadata>-->
      </PUT_METADATA>
    </PUBLISH_METADATA>
  </REQUEST>
</ARCXML>
```

TOC

Used in: RESPONSE
Servers: ArcMap
Parent elements: FCLASS

<TOC >
No Attributes

(m) <TOCGROUP... />

</TOC >

(m): Child element can be used multiple times.

Description:

Main element for specifying a table of contents used with the ArcMap Server in ArcIMS Java Viewers, ArcExplorer 4, and ArcMap.

Restrictions:

- Valid only with the ArcMap Server when using a legend in the ArcIMS Java Viewers, ArcExplorer 4, and ArcMap. Legend information for the ArcIMS HTML Viewer and custom HTML Viewers is handled using LEGEND.

Notes:

None

Examples for TOC:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US"/>
        <UIFONT name="Arial" color="0,0,0" size="12"
style="regular"/><SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576"/>
        <CAPABILITIES forbidden=""/>
      </ENVIRONMENT>
      <LAYOUTINFO pageunits="inches">
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
      </LAYOUTINFO>
      <PROPERTIES>
        <FEATURECOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
 American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
```

```
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
        <FILTERCOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
        <MAPUNITS units="decimal_degrees"/>
        <BACKGROUND color="255,255,255"/>
        <ENVELOPE minx="-198" miny="-162.48914963359" maxx="198"
maxy="156.08518845195" name="Initial_Extent" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" name="CNTRY94" id="0"
visible="true">
        <FCLASS type="polygon"></FCLASS>
        <TOC>
          <TOCGROUP heading="AREA">
            <TOCCLASS label="1041.095 - 364692.187"
description="">Qk0SAAAbm5uEMwg...A</TOCCLASS>
            <TOCCLASS label="364692.188 - 1216700.056"
description="">Qk0SCQAAA...AA¬
```

```
 </TOCCLASS
            <TOCCLASS label="1216700.057 - 6400657.079"
description="">Qk0SCQAA...AA¬
```

```
 </TOCCLASS
          </TOCGROUP>
        </TOC>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

TOCCLASS

Used in: RESPONSE
Servers: ArcMap
Parent elements: TOCGROUP

```
<TOCCLASS

    description ="string"
    label ="string"
>

    No Child Elements
</TOCCLASS >
```

Description:

Provides information about each swatch in a layer along with the label for use with a table of contents.

Restrictions:

- Valid only with the ArcMap Server when using a legend in the ArcIMS Java Viewers, ArcExplorer 4, and ArcMap. Legend information for the ArcIMS HTML Viewer and custom HTML Viewers is handled using LEGEND.

Notes:

- Images embedded inside of TOCCLASS are Base-64-encoded and are not compressed. Note that many images can make the response lengthy.

Attribute Descriptions for TOCCLASS:

Attribute	Usage
description	Text description of the label. This information can be used by a programmer for providing additional information about the label. One example might be tooltips.
label	Label identifying what the TOC image represents.

Examples for TOCCLASS:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
```

```
        <LOCALE language="en" country="US"/>
        <UIFONT name="Arial" color="0,0,0" size="12"
style="regular"/><SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
        <IMAGELIMIT pixelcount="1048576"/>
        <CAPABILITIES forbidden=""/>
      </ENVIRONMENT>
      <LAYOUTINFO pageunits="inches">
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
      </LAYOUTINFO>
      <PROPERTIES>
        <FEATURECOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
        <FILTERCOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
        <MAPUNITS units="decimal_degrees"/>
        <BACKGROUND color="255,255,255"/>
        <ENVELOPE minx="-198" miny="-162.48914963359" maxx="198"
maxy="156.08518845195" name="Initial_Extent" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" name="CNTRY94" id="0"
visible="true">
        <FCLASS type="polygon"></FCLASS>
        <TOC>
          <TOCGROUP heading="AREA">
            <TOCCLASS label="1041.095 - 364692.187"
description="">Qk0SAAAbm5uEMwg...A¬
```

```
 < TOCCLASS>
            <TOCCLASS label="364692.188 - 1216700.056"
description="">Qk0SCQAAA...AA¬
```

```
 < TOCCLASS>
            <TOCCLASS label="1216700.057 - 6400657.079"
description="">Qk0SCQAA...AA¬
```

```
 < TOCCLASS>
          </TOCGROUP>
        </TOC>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

TOCGROUP

Used in: **RESPONSE**
Servers: ArcMap
Parent elements: **TOC**

<TOCGROUP

heading ="*string*"
>

(m) <TOCCLASS... />

</TOCGROUP >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

The main element for representing a layer in a table of contents.

Restrictions:

- Valid only with the ArcMap Server when using a legend in the ArcIMS Java Viewers, ArcExplorer 4, and ArcMap. Legend information for the ArcIMS HTML Viewer and custom HTML Viewers is handled using LEGEND.

Notes:

None

Attribute Descriptions for TOCGROUP:

Attribute	Usage
heading	The overall header for a layer. Only used when there are multiple symbols representing the same layer.

Examples for TOCGROUP:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <ENVIRONMENT>
        <LOCALE language="en" country="US"/>
        <UIFONT name="Arial" color="0,0,0" size="12"
style="regular"/><SEPARATORS cs=" " ts=";"/>
        <SCREEN dpi="96"/>
```

```
        <IMAGELIMIT pixelcount="1048576"/>
        <CAPABILITIES forbidden=""/>
      </ENVIRONMENT>
      <LAYOUTINFO pageunits="inches">
        <ENVELOPE minx="0" miny="0" maxx="8.5" maxy="11" />
      </LAYOUTINFO>
      <PROPERTIES>
        <FEATURECOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
        <FILTERCOORDSYS
string="GEOGCS[&quot;GCS_Assumed_Geographic_1&quot;,DATUM[&quot;D_North
_American_1927&quot;,SPHEROID[&quot;Clarke_1866&quot;,6378206.4,294.978
6982]],PRIMEM[&quot;Greenwich&quot;,0.0],UNIT[&quot;Degree&quot;,0.0174
532925199433]]"/>
        <MAPUNITS units="decimal_degrees"/>
        <BACKGROUND color="255,255,255"/>
        <ENVELOPE minx="-198" miny="-162.48914963359" maxx="198"
maxy="156.08518845195" name="Initial_Extent" />
      </PROPERTIES>
      <LAYERINFO type="featureclass" name="CNTRY94" id="0"
visible="true">
        <FCLASS type="polygon"></FCLASS>
        <TOC>
          <TOCGROUP heading="AREA">
            <TOCCLASS label="1041.095 - 364692.187"
description="">Qk0SAAAbm5uEMwg...A¬
```

```
 </TOCCLASS
            <TOCCLASS label="364692.188 - 1216700.056"
description="">Qk0SCQAAA...AA¬
```

```
 </TOCCLASS
            <TOCCLASS label="1216700.057 - 6400657.079"
description="">Qk0SCQAA...AA¬
```

```
 </TOCCLASS
          </TOCGROUP>
        </TOC>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

TRUETYPEMARKERSYMBOL

Used in: CONFIG REQUEST RESPONSE
Servers: Image Feature ArcMap
Parent elements: EXACT LINE OBJECT OTHER POINT POLYGON RANGE SIMPLERENDERER

<TRUETYPEMARKERSYMBOL

When using ArcMap Server:
character="32 - 255"
angle ="0.0 - 360.0" **[0]**
font ="Any system font" **[Arial]**
fontcolor ="0,0,0 - 255,255,255" **[0,0,0]**
fontsize ="1 - NNN" **[12]**
fontstyle ="regular | bold | italic | underline | outline | bolditalic" **[regular]**
outline ="0,0,0 - 255,255,255"

When using Image or Feature Server:
character="32 - 255"
angle ="0.0 - 360.0" **[0]**
anglefield ="0.0 - 360.0" **[0]**
antialiasing ="true | false" **[false]**
font ="Any system font" **[Arial]**
fontcolor ="0,0,0 - 255,255,255" **[0,0,0]**
fontsize ="1 - NNN" **[12]**
fontstyle ="regular | bold | italic | underline | outline | bolditalic" **[regular]**
glowing ="0,0,0 - 255,255,255"
outline ="0,0,0 - 255,255,255"
overlap ="true | false" **[true]**
rotatemethod ="geographic | arithmetic | mod_arithmetic" **[mod_arithmetic]**
shadow ="0,0,0 - 255,255,255"
transparency ="0.0 - 1.0" **[1.0]**
usecentroid ="true | false" **[true]**
>

No Child Elements
</TRUETYPEMARKERSYMBOL >

Bold: Attribute or child element is required.

Description:

Symbolizes point features using TrueType symbols.

Restrictions:

- *Overlap* is valid only for Image Services. It is ignored when using Feature Services.
- *Outline* and *glowing* should not be used together; use one or the other.
- The attribute *rotatemethod* values "geographic" and "arithmetic" are valid only with Image Services. Feature Services and the legend in ArcExplorer 4 or the ArcIMS Java Viewers support only "mod_arithmetic" for *rotatemethod*.
- In ArcMap Image Services, TRUETYPEMARKERSYMBOL is valid only in acetate layers.
- **Limitation using *angle* and *anglefield* with Image and Feature Services:** When rotating truetype marker symbols in an Image or Feature Service, the marker may not be centered directly on top of the point it represents. The displacement varies depending on the font and character used. Not all characters are affected even within the same font. This issue is not a problem with ArcMap Image Services.

Notes:

- TRUETYPEMARKERSYMBOL is not available from the ArcIMS Author interface.
- The angle for a symbol can be user-defined with *angle*. Angles can also be read from a field in a database table using *anglefield*.
- If the attributes *angle* and *anglefield* are both present, *angle* takes precedence, and *anglefield* is ignored.
- If the symbols are rotating in an unexpected direction, double-check the value for *rotatemethod*.
- If you are using custom TrueType symbols on a UNIX machine, the font must be installed in the following directory:
    - For Solaris: /usr/openwin/lib/X11/fonts/TrueType
    - For AIX: /usr/lpp/X11/lib/X11/fonts/TrueType
    - For Linux: /usr/lib/X11/fonts/TrueType
    - For HP-UX: /usr/lib/X11/fonts/ms.st/typefaces

Attribute Descriptions for TRUETYPEMARKERSYMBOL:

Attribute	Usage
angle	Angle of rotation in degrees.
anglefield	The field in the database that contains the angle of rotation for a TRUETYPEMARKERSYMBOL. The field can be in the layer table or in a joined table. • For shapefiles with no joined tables, the field can be referenced using the short format. field="AREA"

	• For shapefiles with joined tables, the name of the joined table must be included along with the field. field="JOINEDTABLE.AREA" Files with joined fields cannot be read locally in ArcIMS Author or ArcExplorer 4. • For ArcSDE layers without joined tables, the field can be referenced using the short format. field="AREA" The full long name can also be used, but when a long name is used, the file cannot be read locally in ArcIMS Author or ArcExplorer 4. field="ARCSDENAME.TABLE.AREA" • For ArcSDE layers with joined tables, joined fields must be referenced using the full long format. field="ARCSDENAME.TABLE.AREA" Files with joined fields cannot be read locally in ArcIMS Author or ArcExplorer 4. If both *angle* and *anglefield* are used, the attribute *angle* takes precedence.
antialiasing	Used to make edges of labels and symbols smoother. When set to "true", antialiasing is active. Note that the time to generate a map may significantly increase when antialiasing is turned on.
character	Text character ASCII value. The character must be a value between 32 and 255 in a font's character map; characters 0-31 are nonprintable and cannot be used.
font	Font name. The name is case sensitive. If font name uses "&", use "&" instead. For example, ESRI Transportation & Civic should be written as ESRI Transportation & Civic. For Feature Services, the font must reside on the client machine or else the system default font is used.
fontcolor	Font color using RGB values.
fontsize	Font size.
fontstyle	Font style.
glowing	Glow color around symbol using RGB values.
outline	Outline color using RGB values.
overlap	Determines if labels can overlap this symbol. When "true", labels can overlap. When "false", labels will not overlap the symbol. If labels are not drawing as expected, check if *overlap* is set to "false" for this symbol or any other symbol in the ArcIMS service.
rotatemethod	Three methods of calculating angles are available and apply to both *angle* and *anglefield*: 1. "geographic": An angle of 0 is north, and angles are calculated

	clockwise from north. 2. "arithmetic": An angle of 0 is east, and angles are calculated counterclockwise from east. 3. "mod_arithmetic": An angle of 0 is north, and angles are calculated counterclockwise from north.
shadow	Shadow color using RGB values.
transparency	Value to set percentage of transparency. 1.0 is 0 percent transparent. 0.0 is 100 percent transparent.
usecentroid	By default, a marker symbol used on polygon layers draws markers at all polygon vertices. If *usecentroid* is "true", marker is placed in the centroid of the polygon. If multiple polygon parts exist, the marker falls on the part with the biggest area.

Examples for TRUETYPEMARKERSYMBOL:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CITIES" visible="true" id="2">
        <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <TRUETYPEMARKERSYMBOL transparency="0.5"
glowing="0,255,255" shadow="0,0,0" font="ESRI Cartography"
fontstyle="bolditalic" character="252" fontcolor="255,255,0"
fontsize="16" angle="90" antialiasing="false" overlap="true" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: Using data from anglefield to determine the angle rotation.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
```

```
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.21502685546875" miny="18.924781799316406"
maxx="-66.9698486328125" maxy="71.40664672851562" name="Initial_Extent"
/>
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="Cities" visible="true" id="1">
        <DATASET name="cities_rotate" type="point" workspace="shp_ws-0"
/>
        <SIMPLERENDERER>
          <TRUETYPEMARKERSYMBOL transparency="1.0" font="ESRI
Cartography" fontstyle="bolditalic" character="252"
fontcolor="255,255,0" fontsize="30" anglefield="rotate"
rotatemethod="geographic" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 3: When using usecentroid with a polygon layer.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="10">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <SIMPLERENDERER>
            <SIMPLEPOLYGONSYMBOL filltransparency="1.0"
```

```
filltype="solid" fillcolor="0,227,0" />
          </SIMPLERENDERER>
          <SIMPLERENDERER >
            <TRUETYPEMARKERSYMBOL usecentroid="true"
transparency="1.0" font="ESRI Cartography" fontstyle="bold"
character="252" fontcolor="255,255,0" fontsize="24" />
          </SIMPLERENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

UIFONT

Used in: CONFIG RESPONSE Application Server RESPONSE
Parent elements: ENVIRONMENT

```
<UIFONT

    name ="Any system font"
    color ="0,0,0 - 255,255,255"
    size ="1 - NNN"
    style ="regular | bold | italic | underline | outline | bolditalic"
>

    No Child Elements
</UIFONT >
```

Bold: Attribute or child element is required.

Description:

Sets the default font.

Restrictions:

None

Notes:

None

Attribute Descriptions for UIFONT:

Attribute	Usage
color	Font color using RGB values.
name	Font name. The name is case sensitive. If font name uses "&", use "&" instead. For example, ESRI Transportation & Civic should be written as ESRI Transportation & Civic. For Feature Services, the font must reside on the client machine, or else the system default font is used.
size	Font size.
style	Font style.

Examples for UIFONT:

Example 1: When in an application server RESPONSE to GETCLIENTSERVICES.

```
<?xml version="1.0"?>
<ARCXML version="1.1">
<RESPONSE>
  <SERVICES>
    <SERVICE name="world_image" servicegroup="ImageServer1"
access="PUBLIC" type="ImageServer" desc=""  status="ENABLED" >
      <ENVIRONMENT>
        <LOCALE country="US" language="en" />
        <UIFONT name="Arial" />
      </ENVIRONMENT>
      <CLEANUP interval="20" />
    </SERVICE>
  </SERVICES>
</RESPONSE>
</ARCXML>
```

Example 2: When in CONFIG.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180,0" miny="-90,0" maxx="180,0" maxy="90,0"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-18" directory="<path to WORLD
ESRIDATA>" />
      </WORKSPACES>
      <LAYER type="featureclass" name="CNTRY94" visible="true" id="0">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-18" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL fillcolor="127,227,127" filltype="solid"
/>
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

UPDATED

Used in: REQUEST
Servers: Metadata (Browse)
Parent elements: SEARCH_METADATA

<UPDATED

after ="*string*"
before ="*string*"
>

No Child Elements
</UPDATED >

Description:

Allows metadata searches based on a date.

Restrictions:

- At least one attribute must be specified. Both can be used.

Notes:

- The format for dates is YYYY-MM-DD hh:mm:ss. The year, month, and date are required. Hours, minutes, and seconds are optional.

Attribute Descriptions for UPDATED:

Attribute	Usage
after	Search for documents newer than the date and time given.
before	Search for documents older than the date and time given.

Examples for UPDATED:

Example 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_METADATA>
      <SEARCH_METADATA>
        <UPDATED after="2002-1-14"/>
      </SEARCH_METADATA>
    </GET_METADATA>
  </REQUEST>
</ARCXML>
```

USER

Used in: **RESPONSE**
Servers: Metadata (Browse)
Parent elements: **METADATA**

<USER

name ="*string*"
url ="*string*"
>

No Child Elements
</USER >

Bold: Attribute or child element is required.

Description:

Lists user name information.

Restrictions:

- Not available from ArcCatalog.

Notes:

- See GET_USER for request.

Attribute Descriptions for USER:

Attribute	Usage
name	Name of user.
url	URL of Metadata Server's output XML file containing information about the user.

Examples for USER:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <METADATA numresults="1" startresult="1" total="1" >
      <USER name="aUser"
url="http://www.esri.com/MetadataUseraUser.xml" />
    </METADATA>
  </RESPONSE>
</ARCXML>
```

UUID

Used in: RESPONSE
Servers: Metadata (Publish)
Parent elements: METADATA

<UUID >
No Attributes

No Child Elements
</UUID >

Description:

Returns a unique ID that can be used for document IDs.

Restrictions:

None

Notes:

- See GET_UUID for request.
- The unique ID is listed between the opening and closing UUID elements. An example ID is {1a80b5bc-ddfc-11d5-b208-9d1880fdf674}.

Examples for UUID:

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <METADATA>
      <UUID>{1a80b5bd-ddfc-11d5-b208-9d1880fdf674}</UUID>
      <UUID>{1a80b5be-ddfc-11d5-b208-9d1880fdf674}</UUID>
      <UUID>{1a80b5bf-ddfc-11d5-b208-9d1880fdf674}</UUID>
      <UUID>{1a80b5c0-ddfc-11d5-b208-9d1880fdf674}</UUID>
      <UUID>{1a80b5c1-ddfc-11d5-b208-9d1880fdf674}</UUID>
    </METADATA>
  </RESPONSE>
</ARCXML>
```

VALUEMAPLABELRENDERER

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Feature
Parent elements: **GROUPRENDERER** **LAYER** **LAYERDEF** **LAYERINFO** **SCALEDEPENDENTRENDERER**

<VALUEMAPLABELRENDERER

labelfield ="***string***"
lookupfield ="***string***"
featureweight ="no_weight | med_weight | high_weight" **[no_weight]**
howmanylabels ="one_label_per_name | one_label_per_shape | one_label_per_part"
labelbufferratio ="*double*"
labelpriorities ="0,0,0,0,0,0,0,0 - 8,8,8,8,8,8,8,8 | LE_PlaceOnTopHorizontal"
[2,2,1,4,5,3,2,4]
labelweight ="no_weight | med_weight | high_weight" **[high_ weight]**
linelabelposition ="See table below for values" **[PlaceAbove]**
rotationalangles ="*string*"
>

(m) **<EXACT... />** *[Or]*
(m) **<RANGE... />** *[Or]*
<OTHER... />

</VALUEMAPLABELRENDERER >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Labels features. A field is specified for labeling features based on criteria in EXACT or RANGE.

Restrictions:

- The attribute *labelpriorities="LE_PlaceOnTopHorizontal"* is limited to Image Services.
- EXACT or RANGE is required. OTHER is optional.
- Not valid with ArcMap Server.

Notes:

- When rotating symbols, the attribute *labelpriorities* always takes precedence over the attribute *rotationalangles*. If you find that your labels are not rotating as expected when using Image Services, remove the *labelpriorities* attribute if it is present.
- If *subfields* is used in SPATIALQUERY or QUERY for a layer, any fields used for labeling must be included in the subfields list.

- For more information on using renderers, see Using ArcXML Renderers.

Attribute Descriptions for VALUEMAPLABELRENDERER:

Attribute	Usage
featureweight	Prioritizes the importance of features. The feature weight determines how important the feature labeled is for the label placement algorithm. If "no_weight" is specified, then the feature has no importance and can be labeled over. If "high_weight" is specified, then the feature has high importance and cannot be labeled over. Giving importance to features increases the complexity of the labeling problem and also the processing time.
howmanylabels	Determines how often a feature is labeled. • "One_label_per_part" labels all parts of a feature in the case of multipart features. For instance, the State of Hawaii consists of several islands, but they are grouped together as one polygon feature. Each island is labeled. • "One_label_per_name" labels once per feature name. When several features share the same name, only one label is shown. For example, if there is more than one polygon with the name "Residential", only one "Residential" polygon is labeled. • "One_label_per_shape" labels once per feature even if there are multiple segments. For example, the group of islands that make up Hawaii are only labeled once. If there are other features with the same name, they are also labeled.
labelbufferratio	Used to set a buffer around the label. When this is set, no labels overlap within the buffer range. The ratio is the fraction of the height or the width of the label rectangle (whichever is smaller) compared to the width of the buffer. A ratio of "0.0" means no buffer. A ratio of "1.0" means that the buffer is twice the size of the label (the label width equals the buffer width). A negative ratio causes the buffer to be smaller than the label. This can be used to allow labels to overlap.
labelfield	Field containing text for labeling features. The field can be in the layer table or in a joined table. • For shapefiles with no joined tables, the field can be referenced using the short format. labelfield="AREA" • For shapefiles with joined tables, the name of the joined table must be included along with the field. labelfield="JOINEDTABLE.AREA" Files with joined fields cannot be read locally in ArcIMS Author or ArcExplorer 4. • For ArcSDE layers without joined tables, the field can be

	referenced using the short format. labelfield="AREA" The full long name can also be used, but when a long name is used, the file cannot be read locally in ArcIMS Author or ArcExplorer 4. labelfield="ARCSDENAME.TABLE.AREA" • For ArcSDE layers with joined tables, joined fields must be referenced using the full long format. labelfield="ARCSDENAME.TABLE.AREA" Files with joined fields cannot be read locally in ArcIMS Author or ArcExplorer 4. Labels can include data concatenated from two or more fields. Each field name is separated by a space: • Using a shapefile with no joined tables. The short format can be used for field names. labelfield="CITY STATE_NAME" • Using a shapefile with joined tables. The name of the joined table must be included along with the field names. labelfield="JOINEDTABLE.CITY JOINEDTABLE.STATE_NAME" Files with joined fields cannot be read locally in ArcIMS Author or ArcExplorer 4. • Using an ArcSDE layer without joined tables. The short format can be used. labelfield="CITY STATE_NAME" The full long name can also be used. labelfield="ARCSDENAME.TABLE.CITY ARCSDENAME.JOINEDTABLE.STATE_NAME" Files with full long names cannot be read locally in ArcIMS Author or ArcExplorer 4. • Using an ArcSDE layer with joined tables. The full long format must be used. labelfield="ARCSDENAME.TABLE.CITY ARCSDENAME.JOINEDTABLE.STATE_NAME" Files with full long names cannot be read locally in ArcIMS Author or ArcExplorer 4.
labelpriorities	Used to determine where to place the label around the point. The attribute accepts different weights for each of eight positions around the point. Each position corresponds to the positions as shown below: 1 2 3 8 **X** 4 7 6 5

	In each position, the user can prioritize the importance of that position from 0 upwards. 0 signifies that the label should not be placed in that position. 1 means that this is an acceptable position for the label, and all higher numbers represent lesser priorities for that position. For example, "1,0,1,0,0,0,0,0" means that only the first and third label positions will be taken into account when labeling. In another example, "1,2,3,0,1,0,0,0" means try to label at the first and fifth position; if not, then put the label at the second position; if not, then put it at the third position; if this is not possible, then don't label it. <VALUEMAPLABELRENDERER lookupfield="TYPE" labelfield="NAME" labelpriorities="1,2,3,0,1,0,0,0"> Another option is to place a label on top of points rather than around points. To do this, use "LE_PlaceOnTopHorizontal" for the label priority. <VALUEMAPLABELRENDERER lookupfield="TYPE" labelfield="NAME" labelpriorities="LE_PlaceOnTopHorizontal">
labelweight	Used to prioritize the importance of labels. The label weight is usually set to "high_weight" since the labels are more important. This can be set lower if the labels are not as important as the feature.
linelabelposition	Determines where on the line to place the label. The following options are available: • PlaceAbove - Place above the line. • PlaceBelow - Place below the line. • PlaceOnTop - Place on the line. • PlaceLeft - Place along the left side of the line (Label follows line and is not perpendicular to the line). • PlaceRight - Place along the right side of the line (Label follows line and is not perpendicular to the line). • PlaceAboveBelow - Place above or below the line. • PlaceLeftRight - Place at either side of the line. • PlaceInLine - Place anywhere on the line. • PlaceParallel - Place parallel to the line. • PlaceOnTopHorizontal - Place label on top of the line but always horizontal. • PlaceAtStartAbove - Place label at the start above the line. • PlaceAtStartOnTop - Place label at the start on top of the line. • PlaceAtStartBelow - Place label at the start below the line. • PlaceAtEndAbove - Place label at the end above the line. • PlaceAtEndOnTop - Place label at the end on top of the line. • PlaceAtEndBelow - Place label at the end below the line. • PlaceEitherEndAbove - Place at either end above the line. • PlaceEitherEndOnTop - Place at either end on top of the line.

- PlaceEitherEndBelow - Place at either end below the line.

<VALUEMAPLABELRENDERER lookupfield="ADMN_CLASS" labelfield="ROUTE" linelabelposition="PlaceOnTop">

lookupfield	Name of field used to specify ranges for RANGE or exact values for EXACT.
rotationalangles	The rotational angles are possible angles that the label can be placed at, relative to the labeled point. By default, labels are always placed horizontally. To rotate a label, a comma-delimited list of up to eight rotational angles can be given and are prioritized from first to last. For example, if the first priority is to place labels at 45 degrees and the second priority is at 30 degrees, the rotational angles attribute would look like this: <VALUEMAPLABELRENDERER lookupfield="TYPE" labelfield="NAME" rotationalangles="45,30"> *Labelpriorities* always take precedence over *rotationalangles*. If you find that your labels are not rotating as expected, remove the *labelpriorities* attribute if it is present. Alternatively, you can set all the *labelpriorities* to "0". <VALUEMAPLABELRENDERER lookupfield="TYPE" labelfield="NAME" rotationalangles="45,30" > or <VALUEMAPLABELRENDERER lookupfield="TYPE" labelfield="NAME" labelpriorities="0,0,0,0,0,0,0,0" rotationalangles="45,30" >

Examples for VALUEMAPLABELRENDERER:

Example 1: Labels for a line layer using EXACT.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.215027" miny="18.924782" maxx="-66.969849"
maxy="71.406647" name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to USA
```

```
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="ROADS" visible="true" id="1">
        <DATASET name="ROADS" type="line" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <VALUEMAPLABELRENDERER lookupfield="ADMN_CLASS"
labelfield="ROUTE" linelabelposition="placeontop"
howmanylabels="one_label_per_shape">
          <EXACT value="US Highway" label="US Highway">
            <SHIELDSYMBOL font="Arial" fontstyle="regular"
fontsize="10" type="usroad" />
          </EXACT>
          <EXACT value="Interstate" label="Interstate">
            <SHIELDSYMBOL antialiasing="true" font="Arial"
fontstyle="regular" fontsize="10" type="interstate" />
          </EXACT>
          <OTHER>
            <TEXTSYMBOL font="Arial" fontstyle="regular" fontsize="10"
/>
          </OTHER>
        </VALUEMAPLABELRENDERER>
        <SIMPLERENDERER>
            <SIMPLELINESYMBOL type="solid" width="2" color="27,127,27"
/>
          </SIMPLERENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: Labels for a point layer using different types of text symbols in RANGE.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="cities" visible="true" id="1">
        <DATASET name="cities" type="point" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <VALUEMAPLABELRENDERER lookupfield="POPULATION"
```

```
labelfield="NAME" labelpriorities="1,0,0,0,0,0,0,0">
          <RANGE lower="0.0" upper="150000.0" label="Smallest">
            <TEXTSYMBOL font="Tahoma" fontstyle="regular" fontsize="10"
/>
          </RANGE>
          <RANGE lower="150001.0" upper="750000.0" label="Medium">
            <TEXTSYMBOL font="Arial" fontstyle="italic" fontsize="12"
glowing="125,125,125" />
          </RANGE>
          <RANGE lower="750001.0" upper="3427180.0" label="Largest">
            <TEXTSYMBOL font="Times New Roman" fontstyle="bolditalic"
fontsize="14" glowing="255,255,0" shadow="0,0,0" />
          </RANGE>
          </VALUEMAPLABELRENDERER>
          <SIMPLERENDERER>
            <SIMPLEMARKERSYMBOL color="51,102,51" width="8" />
          </SIMPLERENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

VALUEMAPRENDERER

Used in: **CONFIG** **REQUEST** **RESPONSE**
Servers: Image Feature
Parent elements: **GROUPRENDERER** **LAYER** **LAYERDEF** **LAYERINFO** **SCALEDEPENDENTRENDERER**

```
<VALUEMAPRENDERER

    lookupfield ="string"
>
    (m) <EXACT... />
    <OTHER... />
    (m) <RANGE... />

</VALUEMAPRENDERER >
```

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Renders features according to the value in a specified field. Based on this field, a value map can be created to classify data. This is useful when different types of data are stored in the same layer.

Restrictions:

- Not valid with ArcMap Server.

Notes:

- Within the same value map, different types of symbols can be used. As an example, if a value map is done on a point layer, some points could be SIMPLEMARKERSYMBOLs, some points could be RASTERMARKERSYMBOLs, and some points could be TRUETYPEMARKERSYMBOLs.
- If the same symbol type is used throughout a value map, different attribute values can be used. For example, with SIMPLEPOLYGONSYMBOL some polygons can be solid while others cross diagonally.
- For more information on using renderers, see Using ArcXML Renderers.

Attribute Descriptions for VALUEMAPRENDERER:

Attribute	Usage
lookupfield	Name of field used to specify ranges for RANGE or exact values for EXACT.

Examples for VALUEMAPRENDERER:

Example 1: When using EXACT.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="Crime" visible="true" id="2">
        <DATASET name="Crime" type="point" workspace="shp_ws-0" />
        <VALUEMAPRENDERER lookupfield="CODE">
        <EXACT value="1" label="Type 1">
          <SIMPLEMARKERSYMBOL color="27,127,27" type="triangle"
width="6" />
        </EXACT>
        <EXACT value="2" label="Type 2">
          <SIMPLEMARKERSYMBOL color="227,27,27" type="circle"
width="10" />
        </EXACT>
        <EXACT value="3" label="Type 3">
          <TRUETYPEMARKERSYMBOL transparency="1.0" glowing="0,255,255"
font="ESRI Cartography" fontstyle="bold" character="252"
fontcolor="255,255,0" fontsize="16" />
        </EXACT>
        <OTHER>
          <SIMPLEMARKERSYMBOL type="square" width="4" />
        </OTHER>
        </VALUEMAPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 2: When using RANGE.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="Countries" visible="true"
id="1">
        <DATASET name="CNTRY94" type="polygon" workspace="shp_ws-0" />
        <VALUEMAPRENDERER lookupfield="AREA">
        <RANGE lower="0.0" upper="1000000.0" label="Small">
          <GRADIENTFILLSYMBOL type="horizontal" startcolor="255,0,0"
finishcolor="0,255,0" overlap="true" boundary="false" />
        </RANGE>
        <RANGE lower="1000000.001" upper="3000000.000" label="Medium">
          <SIMPLEPOLYGONSYMBOL filltype="cross" fillinterval="4"
fillcolor="255,146,0" />
        </RANGE>
        <RANGE lower="3000000.001" upper="10000000" label="Large">
          <SIMPLEPOLYGONSYMBOL filltype="bdiagonal" fillinterval="6"
fillcolor="255,37,0" />
        </RANGE>
        <OTHER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="128,128,128"
/>
        </OTHER>
        </VALUEMAPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

Example 3: When creating a complex symbol for "Freeway".

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
```

```
        <ENVELOPE minx="-73.933" miny="40.795" maxx="-73.911"
maxy="40.811" name="Initial_Extent" />
         <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
         <SHAPEWORKSPACE name="shp_ws-0" directory="<path to data>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="nyc_roads" visible="true" id="4"
maxscale="1:35000">
      <DATASET name="nyc_roads" type="line" workspace="shp_ws-0" />
        <GROUPRENDERER>
          <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
          <EXACT value="Freeway">
            <SIMPLELINESYMBOL type="solid" width="9" captype="square"
jointype="round" color="0,0,0" />
          </EXACT>
          <EXACT value="Street">
            <SIMPLELINESYMBOL type="solid" width="8" captype="square"
jointype="round" color="0,0,255" />
          </EXACT>
          <OTHER>
            <SIMPLELINESYMBOL type="solid" width="1" captype="round"
jointype="round" color="0,0,255" />
          </OTHER>
          </VALUEMAPRENDERER>
          <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
          <EXACT value="Freeway">
            <SIMPLELINESYMBOL type="solid" width="7" captype="square"
jointype="round" color="255,0,0" />
          </EXACT>
          </VALUEMAPRENDERER>
          <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
          <EXACT value="Freeway">
            <SIMPLELINESYMBOL type="solid" width="3" captype="square"
jointype="round" color="0,0,0" />
          </EXACT>
          </VALUEMAPRENDERER>
          <VALUEMAPRENDERER lookupfield="ROAD_TYPE">
          <EXACT value="Freeway">
            <SIMPLELINESYMBOL type="solid" width="1" captype="round"
jointype="round" color="255,255,255" />
          </EXACT>
          </VALUEMAPRENDERER>
        </GROUPRENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

WORD

Used in: **RESPONSE**
Servers: Metadata (Browse)
Parent elements: **COLLECTION_INFO**

<WORD

value ***="string"***
>

No Child Elements
</WORD >

Bold: Attribute or child element is required.

Description:

Word to use in a search.

Restrictions:

- A search word must be at least three characters in length.

Notes:

None

Attribute Descriptions for WORD:

Attribute	Usage
value	Contains a word from the collection.

Examples for WORD:

Example 1:

```
<?xml version="1.0" encoding="UTF8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <METADATA>
      <COLLECTION_INFO>
        <WORD value="data" />
        <WORD value="esri" />
        <WORD value="geographic" />
         ...
      </COLLECTION_INFO>
    </METADATA>
  </RESPONSE>
</ARCXML>
```

WORKSPACES

Used in: **CONFIG** **REQUEST** **MARKUP**
Servers: Image Query Feature Extract Geocode Metadata (Publish)
Parent elements: **GET_IMAGE** **MAP** **MARKUP** **RESET**

<WORKSPACES >
No Attributes

*When parent element is **MAP in a map configuration file, GET_IMAGE, GET_EXTRACT:***
(m) <IMAGEWORKSPACE... />
(m) <SDEWORKSPACE... />
(m) <SHAPEWORKSPACE... />

*When parent element is **MAP in a viewer configuration file:***
(m) <AVIMSWORKSPACE... />
(m) <FEATURESERVERWORKSPACE... />
(m) <IMAGESERVERWORKSPACE... />
(m) <IMAGEWORKSPACE... />
(m) <MOIMSWORKSPACE... />
(m) <SDEWORKSPACE... />
(m) <SHAPEWORKSPACE... />

*When parent element is **MARKUP:***
(m) <FEATURESERVERWORKSPACE... />

*When parent element is **METADATA_CONFIG or RESET used with the Metadata Server:***
<SDEWORKSPACE... />

</WORKSPACES >

(m): Child element can be used multiple times.

Description:

Specifies a collection of workspaces or directories where the data is located.

Restrictions:

- At least one workspace should be defined in the WORKSPACES element, or no map is generated.
- All DATASET layer names referenced throughout the configuration file or request must be included in WORKSPACES.
- The child elements FEATURESERVERWORKSPACE, IMAGESERVERWORKSPACE, MOIMSWORKSPACE, and AVIMSWORKSPACE can only be used in viewer configuration files such as

default.axl from ArcIMS Designer or output from ArcExplorer 4 or the ArcIMS Java Viewers. They cannot be used in a map configuration file.
- Not valid with ArcMap Server.

Notes:

- If a layer does not show up as expected, double-check that the WORKSPACES directory information is correct for that layer.
- When adding dynamic layers in a request with a new workspace, the WORKSPACE for the data location should be included in the map configuration file instead of the request.
- For additional information on using WORKSPACES in configuration files, see Using Configuration Files.

Examples for WORKSPACES:

Example 1: When in CONFIG and REQUEST.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90"
name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="<path to WORLD
ESRIDATA>"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="forests" visible="true" id="0">
        <DATASET name="forests" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL filltype="solid" fillcolor="0,153,255"
/>
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```