ArcGIS® 9

Using ArcIMS® Route Server for Windows®

GIS by ESRI™

Copyright © 2005-2007 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 and 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, ArcView, ArcIMS, and the ESRI globe logo are trademarks of ESRI, registered in the United States and certain other countries; registration is pending in the European Community. ArcExplorer, ArcEditor, ArcInfo, ArcSDE, ArcLogistics, and the ArcGIS logo are trademarks and www.esri.com is a service mark of ESRI. The Netscape N logo is a registered trademark of Netscape Communications Corporation in the United States and other countries. Other companies and products mentioned herein are trademarks or registered trademarks of their respective trademark owners.

Contents

Introducing ArcIMS Route Server 1

What can you do with ArcIMS Route Server? 2
New features for ArcIMS Route Server 9.2 5
Features new in ArcIMS Route Server 9.1.5 8
Features new in ArcIMS Route Server 9.1 13
Tips on learning ArcIMS Route Server 14

Administering ArcIMS Route Server 15

Managing a Route Virtual Server 16
Route Virtual Services 17
Private Virtual Server 18
Building an ArcIMS Route Server client 19
Projection support 20
Optimization of route stops 21
Route barriers 23
European and North American geocoding styles 25
Reverse intersection support 28
Localization 30
Route Server data requirements 31
Route Server performance tuning 32

ArcIMS Route Server ArcXML 33

ArcIMS Route Server ArcXML schema 34
ArcXML conventions 35

Glossary 179

Index 183

Introducing ArcIMS Route Server

1

IN THIS CHAPTER

- **What can you do with ArcIMS Route Server?**
- **New features of ArcIMS Route Server**
- **Tips on learning ArcIMS Route Server**

Welcome to ESRI® ArcIMS® Route Server, an extension that adds nationwide geocoding and routing to ArcIMS. Once you are familiar with ArcIMS, adding the following capabilities using ArcIMS Route Server is easy:

- **Routing**—Point-to-point routes can be calculated between a set of two or more stops. Calculate a path within a neighborhood, across a city, or across the country. You can display the route results as maps and generate a list of driving directions.
- **Drive time**—You can define the area that can be reached in a specified time or distance from a point using the street network.
- **Geocoding—**ArcIMS Route Server provides the ability to locate addresses on your street map. Complete geocoding capabilities are provided including the ability to calculate point locations for addresses and perform reverse geocoding. This is the process of finding an address by pointing at a location on the map.
- **Street mapping—**ArcIMS Route Server is delivered with a specially formatted street database that can be used for mapping at multiple scales.

This book assumes you are familiar with using ArcIMS. You should feel comfortable administering sites and building custom clients that use ArcXML syntax or working with the provided HTML ASP, and ASP.NET samples. If you need to refresh your knowledge of ArcIMS, see the ArcIMS Help, which contains information on using and customizing ArcIMS as well as the ArcXML Programmer's Reference Guide.

What can you do with ArcIMS Route Server?

ArcIMS Route Server allows you to develop ArcIMS Web sites that provide fast and reliable routing on nationwide street data. The datasets available for ArcIMS Route Server come with ready-to-use map configuration files to be used in your routing Web sites. ArcIMS Route Server works with Spatial Data Compressed (SDC) format—a highly compressed, indexed data format developed to optimize routing and address matching in ArcIMS Web sites.

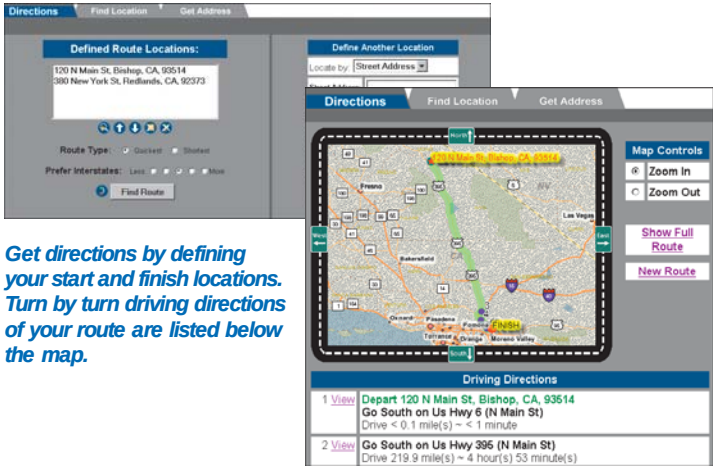
Street mapping at various scales

Complete U.S. national street datasets that allow you to view streets at many scales are available for ArcIMS Route. ArcIMS Route Server comes with samples of each of these street datasets.

Creating a route

When a route is requested, the Route Server processes the beginning and ending addresses entered in the client and returns a route. Route Server can also process multiple stops and allows for setting route preferences.

These results are encoded as an ArcXML structure and consist of a collection of segments, each containing descriptions, turn information, drive-time information, and segment distance as well as the exact location of each segment. Locations are then used by Route Server clients to plot the course of the route on the map. For more information on routing, see GET_ROUTE in Chapter 3, “ArcIMS Route Server ArcXML”.



Get directions by defining your start and finish locations. Turn by turn driving directions of your route are listed below the map.

Geocoding

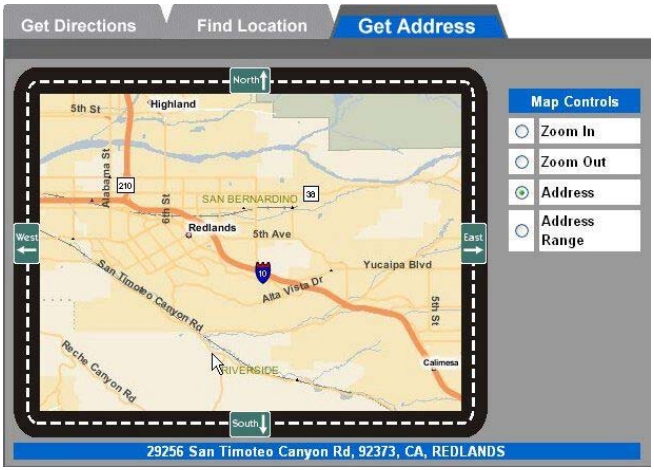
Geocoding is a process of finding the geographic coordinates for an address. Address information, such as street name, number, and postal code, is used to find an x,y location on the map.

Geocoding is based on key components in a street address. A street segment is identified using the postal code or street name and number, city, and state for an address. Location of the address is interpolated along the street segment based on the house number. The side of the street is selected based on an even or odd house number. If more than one address location is found, a list of candidate locations is returned along with a relative score for each.

In this example, an address location is displayed on the map by selecting the Find Location tab, entering address information in the Define Location box, and clicking Locate. If more than one match is found, a list of possible candidates is displayed, and the one with the highest score appears on the map.

Reverse geocoding

Reverse geocoding is a process of identifying the nearest street address for a point on a street segment. ArcIMS Route Server has two styles of reverse geocoding. It can return an address or an address range. For more information on setting up reverse geocoding in ArcIMS Route Server, see GET_GEOCODE in Chapter 3, “ArcIMS Route Server ArcXML”.



Reverse geocoding can help users find an address. In this example, users can click on a street in the map, and an address appears at the bottom of the map window.

New features for ArcIMS Route Server 9.2

Time-based restrictions

Beginning with version 9.2, Route Server supports time-based restrictions for SDC datasets containing time domain information. If the data does not contain time restriction information, this functionality will not be available. An example of a time-based restriction would be an intersection where left turns are prohibited from 4:00 PM to 6:00 PM. To utilize the time-based restriction information, Route Server must be aware of the current time. To this end, a new ArcXML element named TRIP_START has been added to Route Server 9.2, which allows you to specify the departure date and time for a GET_ROUTE request. When this element is defined, the routing engine uses the departure time in conjunction with the accumulated driving time to determine the current time at each stop. It uses this information to determine if a street segment is restricted based on the current time. If it is, a new route is calculated. Note that the TRIP_START time should be specified in the regional time of the departure point. See the TRIP_START element description in Chapter 3 of this document for more details.

Estimated time of arrival (ETA) reporting

The new TRIP_START element also provides the ability to report the estimated arrival times for routes. When TRIP_START is defined, Route Server reports the ETA for each turn in the route. See the SEGMENT element description in Chapter 3 of this document for more details.

Trip planning

Several new ArcXML elements have been added to Route Server that provide trip planning functionality in ArcIMS. Trip planning allows you to schedule multiple-day routes and determine the arrival time at your destination. The primary element is DRIVING_WINDOW, which is used to define the daily driving interval for a route. In other words, you can specify, for example, that driving will start at 9:00 AM and end at 10:00 PM every day of the trip. The route will be split into multiple days if the total trip time is greater than the driving interval. Note that to use the trip planning elements, the TRIP_START element must be defined. See the element descriptions for TRIP_START, DRIVING_DAY, DRIVING_WINDOW, START_TIME, and END_TIME in Chapter 3 of this document for more details.

Additional restrictions enhancements

One-way and turn restrictions on road segments have been exposed in Route Server 9.2 as attribute restrictions, which means they can be enabled and disabled when making a GET_ROUTE request. By default, these restrictions are enabled. To disable them, a new ArcXML element, CLEARRESTRICTION, is available. Disabling one-way and turn restrictions provides the capability to calculate pedestrian and emergency vehicle routes. Note that these restrictions are always “strict” as opposed to being “relaxed”, meaning that when they are enabled they are always applied to the route calculation even if there is no alternative route available to bypass them. The CLEARRESTRICTION element can be used to disable any restriction that is enabled by default, while the SETRESTRCTION element is used to enable restrictions that are disabled by default. Use the GET_SERVICE_INFO request for a report of restrictions available in a dataset and their enabled/disabled status. See the element descriptions for CLEARRESTRICTION and SETRESTRICTION in Chapter 3 of this document for more details.

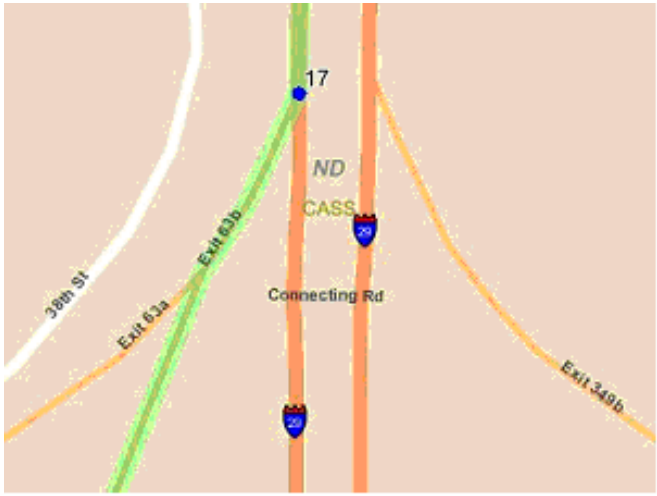
For an example of setting up pedestrian routing, see the following ArcXML snippet:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!—Pedestrian route —>
<ARCXML version="1.1">
 <REQUEST>
   <GET_ROUTE name="">
     <STOPSLIST>
       <ROUTE_STOP id="0" desc="Start">
         <POINT x="-117.16095207735598"
y="34.051969438386266"/>
       </ROUTE_STOP>
       <ROUTE_STOP id="1" desc="Finish">
         <POINT x="-117.15661580451737"
y="34.04469980450941"/>
       </ROUTE_STOP>
     </STOPSLIST>
     <ROUTE_PROPERTIES>
       <RESTRICTIONS>
        <CLEARRESTRICTION name="OneWay" />
         <CLEARRESTRICTION
name="TurnRestriction"  />
       </RESTRICTIONS>
       <ROUTETYPE>quickest</ROUTETYPE>
       <PRECISION>0.0005</PRECISION>
      </ROUTE_PROPERTIES>
    </GET_ROUTE>
  </REQUEST>
</ARCXML>
```

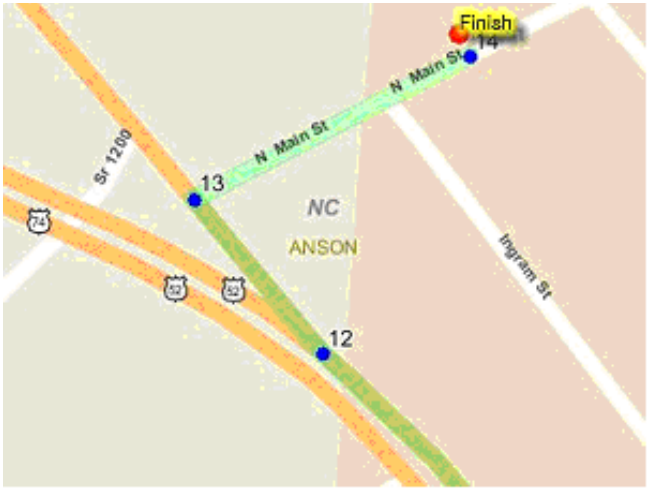
Driving directions improvements

The driving directions engine has been enhanced to provide more detailed driving instructions, including bear left/right and at fork instructions, as well as destination street side information.

1. New *bear left/right* and *at fork keep* driving directions added

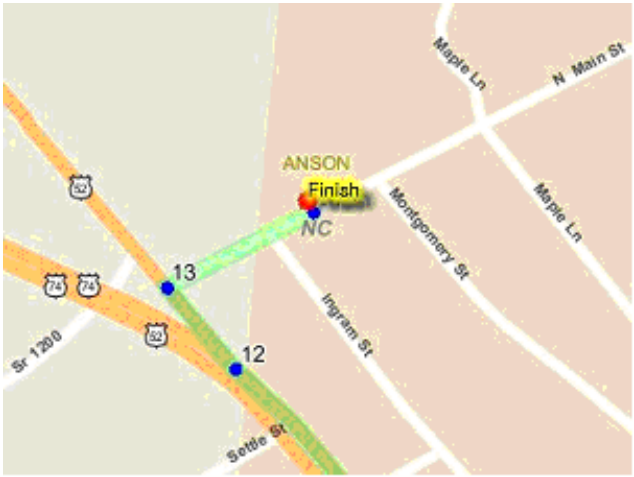


At exit 63B, bear right onto ramp and go on I 94 E toward Fargo



At fork, keep right on US 52 (Salisbury St.)

2. Street side is reported when arriving at a stop.

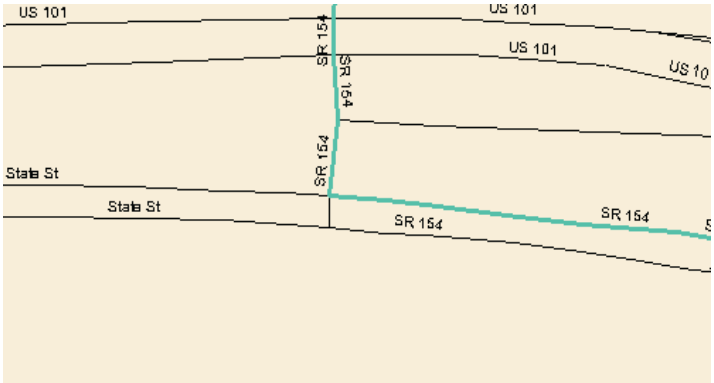


Arrive at 985 N Main St., 28170, WADESBORO, NC, on the left

Directions improvements

1. Report turns at intersections when street name does not change.

This situation occurs when a street turns at an intersection, but the street name does not change. Now there is a Turn instruction generated: Turn right at State St to stay on SR 154.



2. Recognize a U-turn on divided roads.

U-turns on divided roads are now reported as a single U-turn instruction, rather than two left turns.

In the example below, directions will contain the following instruction:

Make U-turn at Ash St. and go back on Brookside Ave

3. Added detail to first direction.

Added ‘toward <intersection street name>’ to the first item in directions for better driver orientation. Example:

Go North on New York St toward State St

Configurable default distance units

The default distance unit for directions output can be set in $ArcIMS\Server\ext\RouteServer\routing.cfg file in [MeasureUnits] section, and will be used in case LENGTH_UNITS tag is not specified in GET_ROUTE request:

```
[MeasureUnits]
UnitsNum = 5
DefaultUnit = 1
[MeasureUnit1]
Name = mile
MileRatio = 1.0
Singular = ::ROUTING.UnitMileSingular
Plural = ::ROUTING.UnitMilePlural
Neutral = ::ROUTING.UnitMileNeutral
```

Configurable U-turn policy

U-turn policy setting can be changed in $ArcIMS\Server\ext\RouteServer\routing.cfg file and will control how solver handles U-turns:

```
[Solver]
BacktrackPolicy = Allowed
```

The following values are accepted:

- Allowed—Allows U-turns unless there is a turn restriction in the data
- Disabled—Prohibits making U-turns anywhere
- DeadendsOnly—Allows U-turns only at dead ends (unless there is a turn restriction in the data)

Restrictions

The goal of using network restrictions is to include parameters such as vehicle type, weight, length, etc, in route calculation for more accurate results. The basic concept is use of a network restriction attribute, which holds a value for each street segment in the network. That value determines whether a segment is passable or not. For example, a restriction attribute can hold TRUE (meaning restricted) for all segments which do not allow trucks to pass. Enabling this restriction for solve will exclude street segments restricted for trucks from route calculation.

Information about available restriction attributes in the routing server is returned as part of SERVICEINFO response. There is a new RESTRICTIONSLIST tag returned as part of EXTENSION tag for the Route extension.

There can be three types of restriction attributes:

- Boolean restrictions (type=boolean)—Can be either true or false for each street segment. If the restriction evaluates true, then the given segment is considered restricted. Example: handling of different vehicle types, such as emergency vehicles or buses, or restricting usage of tollways.
- MaxAllowed restrictions (type=maxallowed)—These contain a floating point value, which represents the maximum value for some parameter. Using this type of restriction requires entering a parameter value before solve. Example: restriction attribute can contain maximum height of a vehicle which can pass a street. When this restriction is used, the user needs to specify vehicle height, so that all edges which have a restriction value less than the parameter value, will be considered restricted.
- MinAllowed restrictions (type=minallowed)—These are the same as MaxAllowed restrictions, but work inthe opposite manner, it contains minimal value for some parameter.

The list includes restriction name and type (boolean, minallowed, maxallowed):

```
<ARCXML version="1.1">
       <RESPONSE>
       <SERVICEINFO>
<LAYERINFO type="featureclass" id="CA_Streets"
name="CA Streets" >
              <FCLASS type="line" />
              <EXTENSION type="Route">
                  <RESTRICTIONSLIST>
<RESTRICTION name="Do Not Use Tollways"
type="boolean" />
<RESTRICTION name="MaxTruckWeight"
type="maxallowed" />
                  </RESTRICTIONSLIST>
              </EXTENSION>
          </LAYERINFO>
       </SERVICEINFO>
       </RESPONSE>
     </ARCXML>
```

It is possible to set restrictions to use in the GET_ROUTE request by adding a RESTRICTIONS tag in the request. Each SETRESTRICTION tag sets a restriction to be used, including name, type and optional parameter for minallowed and maxallowed restrictions.

Restriction usage type can be “strict” or “relaxed”. When restriction is set to be used during solve time, there will be a way to specify whether it should be a “strict” or “relaxed” restriction. Strict restriction means that the solver will not use restricted edges under any circumstances. For example, if there is an island with one bridge and it is restricted, then no route can be found from the island to mainland. Setting a relaxed restriction means that the solver will use restricted edges if there is no alternative route.

```
  <ARCXML version="1.1">
         <REQUEST>
           <GET_ROUTE name="MyRoute">
             <STOPSLIST>
                 ...
             </STOPSLIST>
                 ...
             <ROUTE_PROPERTIES>
               <RESTRICTIONS>
<SETRESTRICTION name="MaxTruckWeight"
type="strict" parameter="5000" />
<SETRESTRICTION name="Do Not Use Tollways"
type="relaxed"/>
               </RESTRICTIONS>
                 ...
             </ROUTE_PROPERTIES>
           </GET_ROUTE>
         </REQUEST>
       </ARCXML>
```

Drivetime

There is a new GET_DRIVETIME type of request, which Route Server can process to generate drive time polygons.

GET_DRIVETIME request contains:

- Center location coordinates
- Network attribute to be used (“shortest” or “quickest”)
- List of drive zone limits
- Optional speed adjustment factor
- Optional coordinate system for request/response (FILTERCOORDSYS and FEATURECOORDSYS)
- Optional direction of the zones being searched in relation to the center location (“inbound” or “outbound”)

DRIVETIME response contains list of calculated drivetime polygons.

Example request:

```
<GET_DRIVETIME>
   <DTCENTER>
     <POINT x=”-96.123" y=”32.97" />
</DTCENTER>
<DTZONES>
   <DTZONEINPUT id=”1" limit=”20" />
   <DTZONEINPUT id=”2" limit=”40" />
   <DTZONEINPUT id=”3" limit=”60" />
</DTZONES>
<DTPROPERTIES>
   <DTTYPE>shortest</DTTYPE>
   <SPEED_FACTOR>1.5</SPEED_FACTOR>
   <DTDIRECTION>outbound</DTDIRECTION>
</DTPROPERTIES>
   <FEATURECOORDSYS id=”4269"/>
   <FILTERCOORDSYS id=”4269"/>
</GET_DRIVETIME>
```

Example response:

```
<DRIVETIME>
   <DTZONERESULT id=”1" limit=”20">
      <POLYGON coords=”-96.123 32.97;-96.32
32.98;…”>
   </DTZONERESULT>
   <DTZONERESULT id=”2" limit=”40">
      <POLYGON coords=”-96.123 32.97;-96.32
32.98;…”>
   </DTZONERESULT>
   <DTZONERESULT id=”3" limit=”60">
      <POLYGON coords=”-96.123 32.97;-96.32
32.98;…”>
   </DTZONERESULT>
</DRIVETIME>
```

Driving speed adjustment

There is a new tag SPEED_FACTOR for the GET_ROUTE and GET_DRIVETIME requests which allows adjustment of driving speeds which are used to calculate estimated driving time. This parameter is a coefficient, which is applied regardless of a road class. Default value is 1. Passing 2 will result in the drive time being calculated using speeds that are twice as fast. A SPEED_FACTOR only affects a route calculated with the 'quickest' option and varies from a value of 0.1 to 10.

Adjusting speeds in GET_ROUTE request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_ROUTE>
      …
      <ROUTE_PROPERTIES>
            …
            <SPEED_FACTOR>1.5</SPEED_FACTOR>
      </ROUTE_PROPERTIES>
    </GET_ROUTE>
  </REQUEST>
</ARCXML>
```

Adjusting speeds in GET_DRIVETIME request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_DRIVETIME>
      …
      <DT_PROPERTIES>
            <SPEED_FACTOR>0.5</SPEED_FACTOR>
      </DT_PROPERTIES>
    </GET_DRIVETIME>
  </REQUEST>
</ARCXML>
```

Search radius

Now it is possible to configure the reverse address geocoder search radius. The search will fail only if no street segments are found inside this area. Otherwise the nearest address will be returned as the result of reverse geocoding.

More countries supported for geocoding

New geocoding styles have been added. The entire list of countries supported and styles available can be found in the "RouteServerGeocodingStyles.pdf" document.

Features new in ArcIMS Route Server 9.1

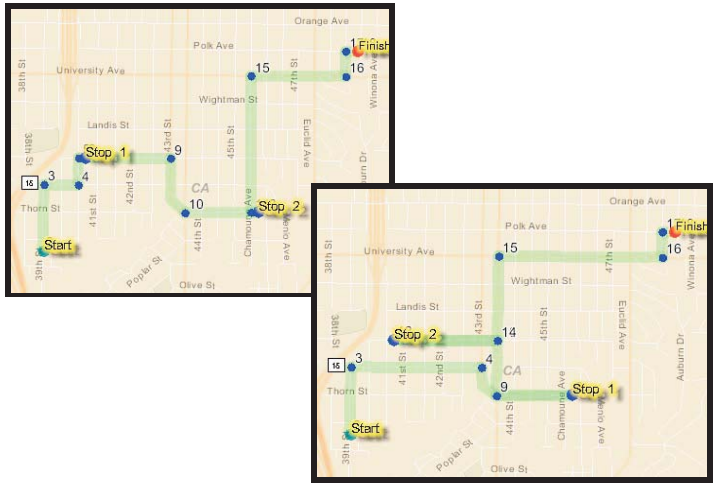
Projection support

ArcIMS Route Server can use any of the projections supported by ArcIMS. This allows you to send map drawing and routing requests with coordinates projected in one coordinate system and get a response in another coordinate system. See Chapter 3, “ArcIMS Route Server ArcXML,” FILTERCOORDSYS and FEATURECOORDSYS for details.

Optimization of route stops

ArcIMS Route Server supports optimization of route stops. Start and Finish stops are fixed, while intermediate stops are reordered to minimize travel distance. (See Chapter 3, “ArcIMS Route Server ArcXML,” OPTIMIZE_STOPS, for details). The optimization feature requires at least two intermediate stops.

The two pictures below illustrate a route with and without optimization of route stops.



Route barriers

ArcIMS Route Server supports both static (specified in GET_ROUTE request) and dynamic route barriers. A barrier is a location on a street segment that prevents a route from continuing along that segment. A barrier can represent obstructions such as road construction, traffic accidents, or traffic congestion. They are useful to help define the most accurate route of travel. Dynamic barriers are stored in a special ArcSDE table, which is located in an ArcSDE database. Route Server periodically polls this table for any barrier updates.

European and North American geocoding styles

Route Server supports both European and North American geocoding styles. A list of supported countries can be found in the *Geocoding Styles* document. New geocoding styles are frequently being developed and added to this list.

Reverse intersection support

In addition to reverse geocoding, ArcIMS Route Server supports reverse intersection geocoding. This allows you to find the intersection nearest a specified location (address or x,y coordinate).

Localization

Localization allows you to generate driving directions in different languages. You can also adjust country-specific attributes, such as decimal separators (dot versus comma), or change between U.S. and metric systems of measurement. See the *Localization Guide* for details on ArcIMS Route Server localization.

Tips on learning ArcIMS Route Server

You should have a basic familiarity with ArcIMS before using ArcIMS Route Server. If you need to refresh your knowledge of ArcIMS, take some time to read *Getting Started With ArcIMS* before installing ArcIMS Route Server. To effectively navigate through ArcIMS Route Server, you need *Installing ArcIMS Route Server* and this book, *Using ArcIMS Route Server*. You can also refer to the *Customizing ArcIMS* documents, which can be downloaded from the topic "ArcIMS general interest and customizing guides" in the ArcIMS help system.

The ArcIMS Knowledge Base

The ArcIMS Knowledge Base is a technical database containing information about ArcIMS and its extension products including ArcIMS Route Server. Use this free online service to find:

- Frequently asked questions
- How-to instructions
- Troubleshooting tips
- Error messages

You can search the Knowledge Base using keywords, or you can browse through folders containing information about different topics related to ArcIMS.

You can access the ArcIMS Knowledge Base from ArcIMS Online at *http://support.esri.com*.

Contacting ESRI

If you need to contact ESRI for technical support, see the support card you received with ArcIMS and ArcIMS Route Server or refer to "Technical support" in the ArcIMS Help system's "Getting help" section.

You can also visit ESRI on the Web at *www.esri.com* for more information on ArcIMS and ArcIMS Route Server.

ESRI education solutions

ESRI provides educational opportunities related to geographic information science, geographic information system (GIS) applications, and technology. You can choose among instructor-led courses, Web-based courses, and self-study workbooks to find education solutions that fit your learning style and budget. For more information, visit *www.esri.com/education* on the Web.

Administering ArcIMS Route Server 2

IN THIS CHAPTER

- Managing a Route Virtual Server
- Building an ArcIMS Route Server client
- Projection Support
- Optimization of Route Stops
- Route Barriers
- European and North American Geocoding Styles
- Reverse Intersection Support
- Localization
- Route Server Data Requirements
- Route Server Performance Tuning

ArcIMS Route Server adds routing, drive time, and additional geocoding functionality to the ArcIMS Spatial Server. When a client sends a routing or geocoding request, the ArcIMS Spatial Server returns the appropriate routing or address information.

This chapter explains how to administer your ArcIMS Route Server and services.

Managing a Route Virtual Server

The ArcIMS Route Server provides the functionality for ArcIMS to perform routing and address matching on SDC-formatted data. This section provides an abbreviated reference to setting up ArcIMS Route Server Services.

To manage ArcIMS Route Services, you should understand the relationship between public and private Virtual Servers used in ArcIMS.

The public Virtual Servers include Image, Feature, ArcMap Image, and Metadata. The services running on these Virtual Servers can be accessed directly through a request. You can start and stop services through one of the administration tools.

The private Virtual Servers include Query, Extract, and Geocode. When you install Route Server, you get two additional private Virtual Servers: Route and SDCGeocode for routing and geocoding, respectively.

Route Virtual Services

Services on Virtual Servers start automatically, depending on information in the map configuration (.axl) file. For Route Services, the map configuration file must include EXTENSION. If you look at one of these files, you will see something similar to the following associated with the streets layer:

```
<EXTENSION type ="Route">
…
</EXTENSION>
<EXTENSION
type="SDCGeocode">
…
</EXTENSION>
```

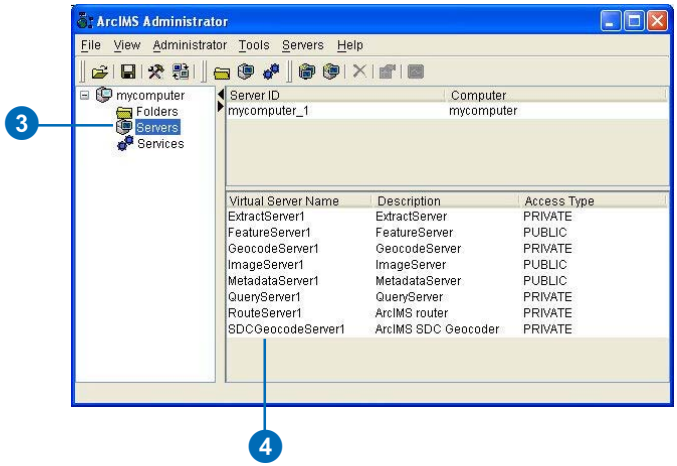
When you start an Image Service that includes these extensions, four other services are started: one on the Image Virtual Server, Route Virtual Server, SDCGeocode Virtual Server, and Query Virtual Server. When you stop the Image Service, all other services are automatically stopped.

See Also

ArcIMS Help topics Managing your ArcIMS Web Site, Managing your site using Administrator or Managing your site using Service Administrator.

Checking Route Virtual Services

1. Click Start > Programs > ArcGIS > ArcIMS > Administrator.
2. Type your username and password in the Login dialog box and click OK.
3. Click Servers from the left panel to open the Servers Manager.
4. Make sure that Route Server and SDCGeocodeServer appear in the servers list. If not, check your Route Server License and contact ESRI for technical support.
5. Click the File menu and click Exit to close the ArcIMS Administrator.



Private Virtual Server

Services running on private Virtual Servers cannot be accessed directly. Instead, a request must be redirected through the service running on the public Virtual Server, in this case an Image Service. The redirection is done in the URL when the request is sent.

See Also

Getting Started With ArcIMS *and ArcIMS Help.*

Accessing private Virtual Servers

- To send a GET_ROUTE or a GET_DRIVETIME request to an Image Service named “myService”, the URL would look similar to this (all one line):

```
http://mycomputer.domain.com/servlet/
com.esri.esrimap.Esrimap?ClientVersion=9.0
&ServiceName=myService&CustomService=Route
&Form=True&Encode=True
```

- A similar URL is used for a GET_GEOCODE request (all one line):

```
http://mycomputer.domain.com/servlet/
com.esri.esrimap.Esrimap?ClientVersion=9.0
&ServiceName=myService&CustomService=SDCGeocode
&Form=True&Encode=True
```

Building an ArcIMS Route Server client

With ArcIMS Route Server, you can use any ArcIMS client that supports an Image Service, ArcXML geocoding, and routing requests. You can build your own custom client or use the HTML sample provided with ArcIMS Route Server. Most of the examples of ArcIMS Route Server shown in this book use this sample. Your ArcIMS Route Server Web site will most likely look quite different. Information on using the ArcIMS Route Server sample viewer is available in the *Sample_Viewer_Setup_and_Configuration.htm* document located in the sample folder. For information on building or customizing your own ArcIMS client, see the *Customizing ArcIMS* documents accessed from the ArcIMS help file topic "ArcIMS general interest and customizing guides".

Once you create Services with the ArcIMS Route Server map configuration file, you are ready to build a Web site. You can build any ArcIMS client that supports Image Services and GET_GEOCODE and GET_ROUTE requests. Web site design needs to be done in a custom environment because routing tools and other related functionality are not available in ArcIMS Designer. ArcIMS Author will not recognize SDC map configuration files. In order to modify SDC map configuration files, you need to use XML or a text editor to manually edit the tags in the ArcXML file.

Projection support

ArcIMS Route Server can use any of the projections supported by ArcIMS. This allows you to send map drawing and routing requests with coordinates projected in one coordinate system, and get a response in another coordinate system. See FILTERCOORDSYS and FEATURECOORDSYS in Chapter 3, “ArcIMS Route Server ArcXML” for details.

Projection support allows you to view a map in the projection you are accustomed to, whether the source data has the same projection or not.

Projections may be specified in several different ways.

- The projection for each SDC data layer is described in the .prj file associated with the .sdc file. The .prj file is required for data layers and as a rule is delivered together with the SDC layer.
- The coordinate system of the requests can be specified in the FILTERCOORDSYS in the ArcXML file. This coordinate system is used as the default for requests when a projection is not specified in the request.
- The coordinate system of all coordinates returned in the ROUTE, GEOCODE, or IMAGE response can be specified in the FEATURECOORDSYS tag of the request. If a coordinate system is not specified in the request, the default is as specified in the ArcXML file.
- In a map configuration file, FILTERCOORDSYS and FEATURECOORDSYS should both be present, and they should have the same value.

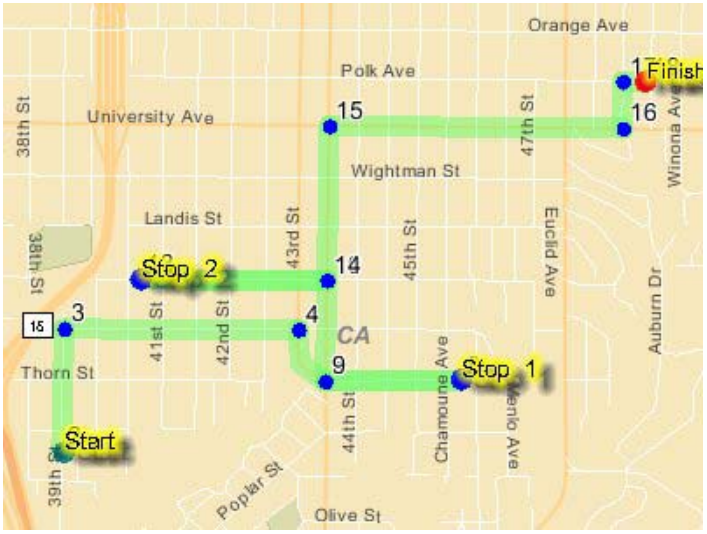
You can specify any of the projections that are supported in ArcIMS, but specifying the datum transformation is not an option.

Known limit—Datum transformation attributes for FILTERCOORDSYS and FEATURECOORDSYS datumtransform ID and datumtransform strings are not supported with RouteServer. If you are using these attributes, the Image Service map and the associated route or geocoded point may not match.

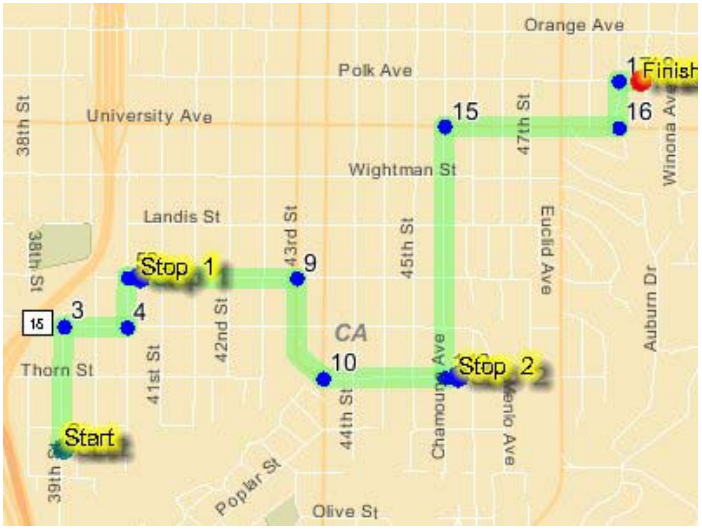
Optimization of route stops

ArcIMS Route Server supports optimization of route stops. Start and Finish stops are fixed, while intermediate stops are reordered to minimize travel distance. See OPTIMIZE_STOPS in Chapter 3, “ArcIMS Route Server ArcXML” for details. The optimization feature requires at least two intermediate stops.

The two pictures below illustrate a nonoptimized and an optimized route.



non-optimized route



optimized route

Optimizing route stops

In order to optimize the order of routing stops, specify OPTIMIZE_STOPS in a GET_ROUTE request. The response will return driving directions in optimized order.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
<REQUEST>
<GET_ROUTE name="Router”>
  <STOPSLIST>
    <ROUTE_STOP id="Stop0" desc="Stop0">
      <POINT x="-81.7302" y="41.4624"/>
    </ROUTE_STOP>
    <ROUTE_STOP id="Stop1" desc="Stop1">
      <POINT x="-81.7183" y="41.4587"/>
    </ROUTE_STOP>
    <ROUTE_STOP id="Stop2" desc="Stop2">
      <POINT x="-81.7302" y="41.4670"/>
    </ROUTE_STOP>
    <ROUTE_STOP id="Stop3" desc="Stop3">
      <POINT x="-81.7149" y="41.4599"/>
    </ROUTE_STOP>
    <ROUTE_STOP id=”Stop4" desc="Stop4">
      <POINT x="-81.7235" y="41.4682"/>
    </ROUTE_STOP>
    <ROUTE_STOP id="Stop5" desc="Stop5">
      <POINT x="-81.7165" y="41.4697"/>
    </ROUTE_STOP>
  </STOPSLIST>
<ROUTE_PROPERTIES>
   <ROUTETYPE>quickest</ROUTETYPE>
   <HWYPREF>50</HWYPREF>
   <PRECISION>0.1463266191428886</PRECISION>
   <LOCALE language="en" country="US" variant=""
/>
     <OPTIMIZE_STOPS enable="true" />
   <LENGTH_UNITS name="kilometer"/>
 </ROUTE_PROPERTIES>
</GET_ROUTE>
</REQUEST>
</ARCXML>
```

Route barriers

A barrier is a location on a street segment that prevents a route from continuing further along that segment. A barrier can represent obstructions such as road construction, traffic accidents, or congestion. They are useful to help define the most accurate route of travel. ArcIMS Route Server supports barriers on a per-request basis, or they can be dynamically fed via a special ArcSDE table.

Using per-request barriers

Below is an example of using a per-request barrier in ArcIMS Route Server with the GET_ROUTE request.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
 <REQUEST>
 <GET_ROUTE name="Route Test for Barriers">
    <STOPSLIST>
   ...
    </STOPSLIST>
  <BARRIERSLIST>
    <BARRIER_POINT>
      <POINT x="-90.212855" y="38.611982"/>
    </BARRIER_POINT>
    <BARRIER_POINT>
      <POINT x="-89.908691" y="38.744964"/>
    </BARRIER_POINT>
  </BARRIERSLIST>
  <ROUTEFILTER>
    <ENVELOPE minx="-180.0063613231552"
miny="11.358778625954183" maxx="-
64.99363867684477" maxy="78.64122137404581"/>
  </ROUTEFILTER>
   <ROUTE_PROPERTIES>
      <ROUTETYPE>shortest</ROUTETYPE>
      <HWYPREF>50</HWYPREF>
      <PRECISION>0.1463266191428886</PRECISION>
  </ROUTE_PROPERTIES>
 </GET_ROUTE>
</REQUEST>
</ARCXML>
```

Dynamic barriers

Another way to specify barriers is to store barrier locations in an ArcSDE table. In this case, barrier locations are polled from the table at specified intervals to dynamically update placed barriers. These dynamically updated barriers are added to any barriers that have been placed with the GET_ROUTE request.

In the ArcSDE table, barriers are represented by a pair of coordinates (must be in latitude and longitude) and by the ID of a street segment (Street ID). A Street ID is optional and used to precisely snap the barrier location to the street network. It is used in cases where there is more than one street candidate for a barrier, such as when there is a barrier at an intersection and it is not clear which road is blocked for routing. The configuration file, aimsrt.cfg, is used for setting parameters for dynamic barriers and is located in the <ArcIMS installation folder>\Server\etc folder. The parameters below are used to configure dynamic barriers.

aimsrt.cfg file parameters

SDEStreetID contains the name of the ArcSDE table field where Street IDs are stored. If the parameter SDEStreetID is present, SDCStreetID is required in the table; if it is omitted, SDCStreetID should be omitted from the table, too.

SDCStreetID contains the name of the source SDC table field where all Street IDs are stored. If the parameter SDCStreetID is present, SDEStreetID is required in the table; if it is omitted, SDEStreetID should be omitted from the table. When a new barrier appears, Route Server compares the value of the SDEStreetID field in the SDC table with the SDCStreetID of the new barrier. If a segment ID matches, the barrier is placed on this segment. If there is no StreetID match, the barrier will be placed on the closest street segment.

SDEBarriersPollTimeout (required) specifies the poll time interval, measured in seconds, that Route Server uses in polling the ArcSDE table. This parameter is set considering both the table size and update frequency. It is not recommended to specify a small value for SDEBarriersPollTimeout, as it does not permit sufficient time to poll the SDE table and may cause server performance degradation. Recommended poll timeout values are five minutes or more.

The steps to implement dynamic barriers in ArcIMS Route Server are as follows:

1. Put barrier locations in an ArcSDE table. The table can reside on any SDE server and should have point geometry and an integer field containing Street ID for each barrier location.
2. Add the barriers table as a layer to the map by adding a new WORKSPACE and LAYER in the map configuration (.axl) file. The layer ID must be RS_DYNAMIC_BARRIERS. For example,

```
<WORKSPACES>
<SDEWORKSPACE name="dbws" instance="port:5151"
server="dragon" user="sde” password="2ghWj.c"
database=""/>
</WORKSPACES>
<LAYER type="featureclass" name="Dynamic SDE
Barriers" visible="false"
id="RS_DYNAMIC_BARRIERS”>
<DATASET name="Barriers" type="point"
workspace="dbws"/>
</LAYER>
```

3. Modify the aimsrt.cfg, to set the **SDEBarriersPollTimeout**, **SDEStreetID**, and **SDCStreetID** parameters.
4. Restart ArcIMS Services.
5. After dynamic barriers are set up, GET_ROUTE requests will trigger Route Server to poll the ArcSDE table for updates of the dynamic barriers.

European and North American geocoding styles

Route Server supports both European and North American geocoding styles. A list of supported countries can be found in the *Geocoding Styles* document. New geocoding styles are frequently developed and added to this list.

It is possible to geocode the following attributes using various styles supported by Route Server including:

- street address
- street intersection
- postal code
- city
- reverse address geocoding
- single field search

Using Route Server geocoding styles

The geocoding styles available for use are determined by the geocoding indexes built during preparation of the SDC dataset. In order to activate a style within a specific SDC dataset, follow these steps.

1. Open map definition file (.axl).
2. Locate each occurrence of a GCSTYLE tag. The name attribute of the tag specifies the name of the available geocoding style. This example defines the street address geocoding style for the US.

```
<EXTENSION type="SDCGeocode">
  <GCSTYLE name="SDC.LINE_STREET_ADDRESS.USA">
      <GCFIELD id="L_F_ADD" name="L_F_ADD"/>
      <GCFIELD id="R_F_ADD" name="R_F_ADD"/>
      <GCFIELD id="L_T_ADD" name="L_T_ADD"/>
      <GCFIELD id="R_T_ADD" name="R_T_ADD"/>
      <GCFIELD id="PREFIX" name="PREFIX"/>
      <GCFIELD id="NAME" name="NAME"/>
      <GCFIELD id="TYPE" name="TYPE"/>
      <GCFIELD id="SUFFIX" name="SUFFIX"/>
    </GCSTYLE>
  </EXTENSION>
```

3. Refer to the *Geocoding Styles* document to find which input fields are supported for that specific style. For example, SDC.LINE_STREET_ADDRESS.USA style has the inputs Street Address, five-digit ZIP Code, City, and State.

   Below is a sample GET GEOCODE request for that style.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
   <GET_GEOCODE maxcandidates="10" minscore="60">
      <LAYER id="18" />
      <ADDRESS>
        <GCTAG id="Street Address" value="380 new
york St" />
        <GCTAG id="5-digit ZIP code"
value="92373" />
        <GCTAG id="City" value="redlands" />
        <GCTAG id="State" value="ca" />
      </ADDRESS>
     </GET_GEOCODE>
  </REQUEST>
</ARCXML>
```

4. Refer to the *Geocoding Styles* document to find the output fields for a specific style. For example, the outputs for the SDC.LINE_STREET_ADDRESS.USA style are:

   • SCORE

   •ADDRESSFOUND

   •SHAPEFIELD

   •House Range

   •Name

   •Postal Code

   •City

   •State or Province

   •Country

   Here is a sample GEOCODE response for that style.

```
<?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="34" precision="0">
          <FIELDVALUE valuestring="380 New York
St, Redlands, CA, 92373" />
       </FIELD>
       <FIELD type="-98" name="SHAPEFIELD”>
          <FIELDVALUE>
             <POINT x="-70.147315455731"
y="44.102606073962" />
          </FIELDVALUE>
       </FIELD>
       <FIELD type="12" name="House Range"
size="3" precision="0">
          <FIELDVALUE valuestring="380" />
       </FIELD>
       <FIELD type="12" name="Name" size="9"
precision="0">
          <FIELDVALUE valuestring="New York St" /
>
       </FIELD>
       <FIELD type="12" name="Postal Code"
size="5" precision="0">
          <FIELDVALUE valuestring="92373" />
       </FIELD>
       <FIELD type=”12" name=”City” size=”8"
precision="0">
          <FIELDVALUE valuestring="Redlands" />
       </FIELD>
       <FIELD type="12" name="State or Province"
size="2" precision="0">
          <FIELDVALUE valuestring=”CA” />
       </FIELD>
       <FIELD type="12" name="Country" size="2"
precision="0">
          <FIELDVALUE valuestring="US" />
       </FIELD>
     </FEATURE>
     <GCCOUNT count="1" />
   </GEOCODE>
</RESPONSE>
</ARCXML>
```

Geocoding parameters

Each geocoding style has a set of supported parameters. The list of parameters for each style can be found in the *Geocoding Styles* document.

For example, SDC.LINE_STREET_ADDRESS.USA style supports the following parameters:

- spelling sesitivity
- minscore
- maxcandidates
- offset

The geocoding offset parameter defines the distance a located point will be offset from the street containing the found address. The default distance is 10 meters. The geocoding offset parameter can be specified in the ArcXML file using the GCPARAM tag. See GCPARAM in Chapter 3, “ArcIMS Route Server ArcXML” for details.

An example of specifying geocoding offset:

```
<LAYER type="featureclass" name="Streets"
visible="true" id="18" maxscale="1:45700">
  <EXTENSION type="SDCGeocode">
      <GCSTYLE
          name="SDC.LINE_ADDRESS_STREETMAP.USA">
        <GCFIELD id="L_F_ADD" name="L_F_ADD"/>
        <GCFIELD id="R_F_ADD" name="R_F_ADD"/>
        <GCFIELD id="L_T_ADD" name="L_T_ADD”/>
        <GCFIELD id="R_T_ADD" name="R_T_ADD”/>
        <GCFIELD id="PREFIX" name="PREFIX"/>
        <GCFIELD id="NAME" name="NAME"/>
        <GCFIELD id="TYPE" name="TYPE"/>
        <GCFIELD id="SUFFIX" name="SUFFIX"/>
            <GCPARAM id=”Offset” value=”50"/>
              </GCSTYLE>
            </EXTENSION>
    </LAYER>
```

The parameters for minimum matching score, maximum number of candidates, and spelling sensitivity can be specified in the GET_GEOCODE tag. See GET_GEOCODE in Chapter 3, ““ArcIMS Route Server ArcXML” for details. For example:

```
<ARCXML version="1.1">
  <REQUEST>
   <GET_GEOCODE maxcandidates="10" minscore="60"
spellingsensitivity="90" >
     <LAYER id="18" />
     <ADDRESS>
       <GCTAG id="Address" value="CARRER CEDRE
4" />
       <GCTAG id="CITY" value="Santa Coloma" />
       <GCTAG id="Country" value="Andorra" />
     </ADDRESS>
   </GET_GEOCODE>
  </REQUEST>
</ARCXML>
```

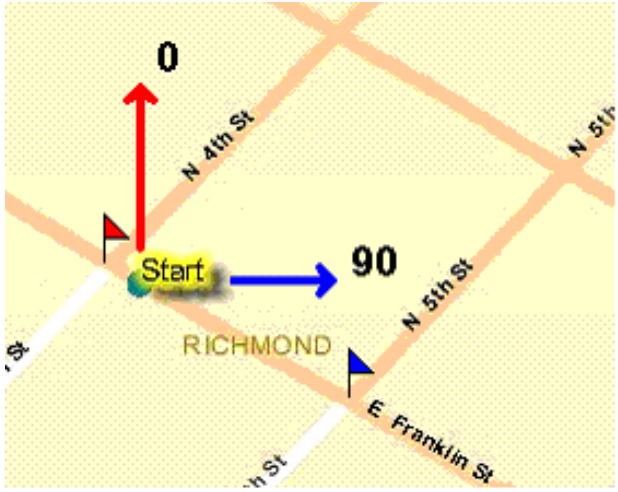
Reverse intersection support

Reverse intersection geocoding finds the closest street intersection to a given location. Once the location is placed on the closest street segment, the street is traversed in both directions until an intersection is found. Any two intersecting streets that have different names are considered an intersection. Each unique pair of different street names found at an intersection is returned as an intersection candidate.

The search can be performed in a specified direction along the street. This can be used when coordinates and direction are read from a GPS device and there is a need to find a closest intersection in the direction of travel.

In the following figure, if you are driving east (azimuth is 90 degrees), the nearest intersection will be E Franklin St & N 5th St, if you are driving north (azimuth is 0 degrees), the nearest intersection will be E Franklin St & N 4th St.

In case direction will not be specified, both E Franklin St & N 4th St and E Franklin St & N 5th St intersections will be returned as results.



Example request to find an intersection regardless of direction:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
   <GET_GEOCODE maxcandidates="25" minscore="60">
    <LAYER id="18" />
    <ADDRESS>
      <GCTAG id="style" value="intersection" />
      <GCTAG id="x" value=" -77.0132459997014" />
      <GCTAG id="y" value=" 38.8869969991112" />
      </ADDRESS>
    </GET_GEOCODE>
  </REQUEST>
</ARCXML>
```

Example request based on finding the closest intersection while traveling south (direction = 180 degrees). **Direction** value should be specified in decimal degrees clockwise, with North being zero azimuth.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
   <GET_GEOCODE maxcandidates="25" minscore="60">
     <LAYER id="18" />
     <ADDRESS>
      <GCTAG id="style" value="intersection" />
      <GCTAG id="direction" value="180" />
      <GCTAG id="x" value=" -77.0132459997014" />
      <GCTAG id="y" value=" 38.8869969991112" />
      </ADDRESS>
    </GET_GEOCODE>
  </REQUEST>
</ARCXML>
```

Localization

Localization allows you to generate driving directions in many languages. New language support can be added by editing the Route Server language configuration file, ROUTING.LNG, located in the \ArcGIS\ArcIMS\Server\ext\RouteServer\ folder. See the *Localization Guide* for details on Route Server localization.

You can also adjust country-specific attributes, such as decimal separators (dot versus comma), or change between U.S. and metric systems of measurement.

The language for driving directions is defined in the LOCALE tag of the GET_ROUTE request. For details on LOCALE tag usage see LOCALE in Chapter 3, “ArcIMS Route Server ArcXML” . Here is an example of specifying the German language in a GET_ROUTE request:

```
<ARCXML version="1.1">
  <REQUEST>
    <GET_ROUTE name="myroute”>
      <STOPSLIST>
        <ROUTE_STOP id="0" desc="75080">
          <POINT x="-96.743072" y="32.971216"/>
        </ROUTE_STOP>
        <ROUTE_STOP id="1" desc="75081">
          <POINT x="-96.710713" y="32.948611"/>
        </ROUTE_STOP>
      </STOPSLIST>
      <ROUTEFILTER>
        <ENVELOPE minx="-180.0063613231552"
miny="11.358778625954183" maxx="-
64.99363867684477" maxy="78.64122137404581"/>
      </ROUTEFILTER>
      <ROUTE_PROPERTIES>
        <ROUTETYPE>quickest</ROUTETYPE>
        <HWYPREF>50</HWYPREF>
        <PRECISION>0.1463266191428886</PRECISION>
        <LOCALE language="de" country="DE" vari-
ant="" />
       <LENGTH_UNITS name="kilometer"/>
</ROUTE_PROPERTIES>
    </GET_ROUTE>
  </REQUEST>
</ARCXML>
```

ArcIMS requires that requests use character sets compatible with the one set in the server. Route Server must comply with this by the LOCALE tag specifying a compatible character set to the LOCALE set in the server. Otherwise, ArcIMS will fail to correctly parse the request.

ArcIMS requires you to set which Locale you are running.

Route Server data requirements

Data requirements

Route Server requires SDC 2 datasets.

If you have routing indexes for any layer and you want to have routing for this layer, add an EXTENSION tag with the Route attribute type.

```
<LAYER type="featureclass" name="Streets"
visible="true" id="4">
  <DATASET name="streets" type="line"
workspace="sdc_ws" />
  <EXTENSION type="Route" />
```

If you have geocoding indexes for any layer and you want to have geocoding for this layer, then add an EXTENSION tag with the SDCGeocode attribute type.

```
<LAYER type="featureclass" name="Streets"
visible="true" id="4">
  <DATASET name="geocode" type="line"
workspace="sdc_ws" />
  <EXTENSION type="SDCGeocode" />
```

Note, that routing and geocoding extensions may be assigned to one layer.

```
<LAYER type="featureclass" name="Streets"
visible="true" id="4">
  <DATASET name="streets" type="line”
workspace="sdc_ws" />
  <EXTENSION type="Route" />
  <EXTENSION type="SDCGeocode" />
```

SDC data format

Smart Data Compression is an encrypted and highly compressed data format. This format is used by commercial data vendors who distribute street data for geocoding and routing with various ESRI software products.

Creating Route Server datasets

The Data Development Kit (DDK Pro) toolkit is available for limited commercial use to create National datasets. Dataset creation requires optimization of the data for acceptable performance.

Available datasets and included samples

North American and European national datasets are available for Route Server for nationwide routing. A few samples are provided with Route Server.

Route Server performance tuning

The following parameters of the configuration file *(*aimsrt.cfg), located in folder <ArcIMS installation folder>\Server\etc, can affect ArcIMS Route Server performance. Some examples are provided below.

- **CacheType** (required)—Specifies where RouteServer cache results will be stored (in memory or on disk). It can have the values of disk or memory. If the value is disk, then the CacheDir parameter is required. For example, setting a larger route cache size may reduce user wait time when zooming in or out.
- **CacheDir** (required if CacheType has a value of disk, otherwise it should be omitted.)—CacheDir sets the absolute local path for the cache storage folder. In the case of multiple spatial servers running Route Server, it is recommended that the directory of the disk cache and map cache be the same as the UNC location for all spatial servers. This should be done so route and sequential map draw requests reuse the results that are stored in a shared disk cache. Disk cache can also be used to reduce memory consumption on the server.

  In case of a single spatial server running Route Server it is recommended to use memory cache for better performance.
- **MaxCachedRoutes** (required)—Maximum cache size indicates how many results (routes) will be kept in the cache. On average, one route requires between 50K to 1 MB. The number of expected concurrent users requesting routes must also be taken into consideration when determining the size of the cache. Route Server comes with a default setting of a maximum of five routes in cache. It is recommended to estimate the peak number of concurrent users requesting a route and ensure MaxCachedRoutes exceeds that estimate.

ArcIMS Route Server ArcXML

3

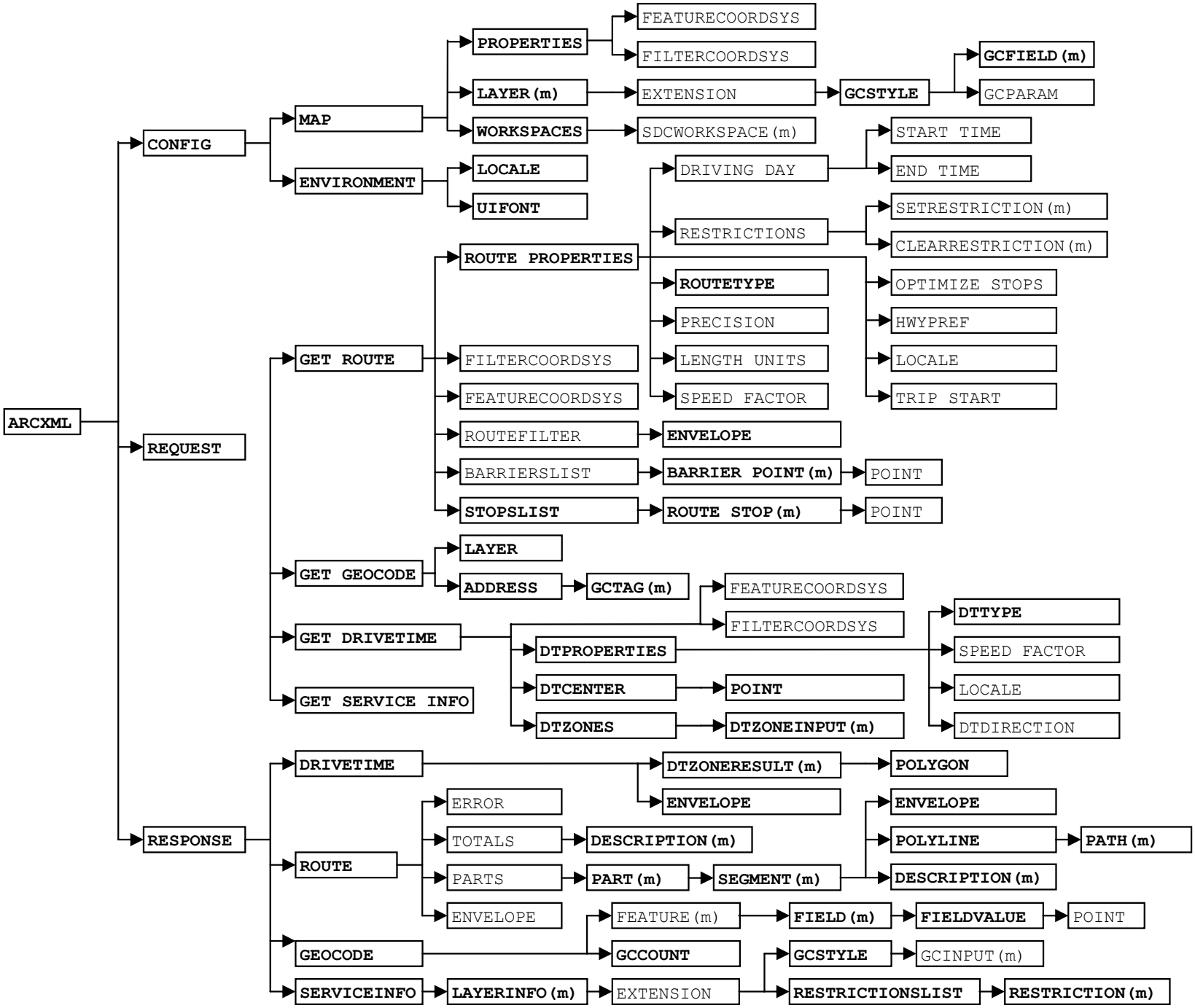
IN THIS CHAPTER

- **ArcIMS Route Server ArcXML schema**
- **ArcXML conventions**
- **ArcIMS Element Reference**

ArcIMS Route Server, uses ArcXML to communicate with the ArcIMS Spatial Server. It uses many of the same ArcXML elements along with additions to communicate routing and address matching.

This chapter provides information on elements used by ArcIMS Route Server. It assumes you are familiar with using ArcXML. For a complete listing of ArcXML elements and more information on working with ArcXML, see *ArcXML Programmer's Reference Guide*.

ArcIMS Route Server ArcXML schema



ArcXML conventions

The ArcXML element structure includes the element name and any attributes and child elements. This page outlines the conventions used in documenting ArcXML.

- Elements are always written in uppercase letters.
- Attributes are always written in lowercase letters. Bold means the attribute is required.
- Definitions of attributes are found in the attribute table for the element. If an attribute has a defined list of valid values, all possible values are listed, but note that only one value can be used at a time. Attribute values are always placed inside double quotes (“”), and the text is not case sensitive unless noted. If the attribute has a default value, it is listed in brackets after the attribute list or type. If an attribute does not have a list of known values, the value type is listed, such as double, integer, or string.
- Child elements are always written in uppercase letters. If the child element is required, it is bold. Child elements that can be used multiple are denoted by an m in parenthesis (m). In some cases, special instructions are given in brackets after the element. Special instructions are given in the ‘Restrictions’ section of each element. The most common scenario is when a group of child elements is listed but only one child element can be used.

Contents

ADDRESS 40
ARCXML 42
BARRIERSLIST 44
BARRIER_POINT 46
CLEARRESTRICTION 48
COORDS 50
DESCRIPTION 52
DRIVETIME 56
DRIVING_DAY 58
DRIVING_WINDOW 60
DTCENTER 62
DTDIRECTION 64
DTPROPERTIES 66
DTTYPE 68
DTZONEINPUT 70
DTZONERESULT 72
DTZONES 74
END_TIME 76
ENVELOPE 78
ERROR 82
EXTENSION 85
FEATURE 89
FEATURECOORDSYS 91
FIELD 93
FIELDVALUE 96

FILTERCOORDSYS 98
GCCOUNT 101
GCINPUT 103
GCPARAM 105
GCSTYLE 107
GCTAG 110
GEOCODE 113
GET_DRIVETIME 115
GET_GEOCODE 117
GET_ROUTE 119
GET_SERVICE_INFO 121
HWYPREF 122
LAYER 124
LAYERINFO 128
LENGTH_UNITS 130
LOCALE 132
OPTIMIZE_STOPS 135
PART 137
PARTS 139
PATH 141
POINT 143
POLYGON 146
POLYLINE 148
PRECISION 150
REQUEST 152

RESPONSE 154
RESTRICTION 156
RESTRICTIONS 158
RESTRICTIONSLIST 160
ROUTE 162
ROUTE_PROPERTIES 165
ROUTE_STOP 167
ROUTEFILTER 169
ROUTETYPE 171
SDCWORKSPACE 173
SEGMENT 175
SERVICEINFO 180
SETRESTRICTION 182
SPEED_FACTOR 185
START_TIME 188
STOPSLIST 190
TOTALS 192
TRIP_START 194

ADDRESS

Used in: REQUEST
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:

None

Example for ADDRESS:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_GEOCODE maxcandidates="20" minscore="60" >
      <LAYER id="18" />
        <ADDRESS>
          <GCTAG id="Street Address" value="380 new york St" />
```

```
          <GCTAG id="5-digit ZIP Code" value="92373" />
          <GCTAG id="City" value="redlands" />
          <GCTAG id="State" value="ca" />
        </ADDRESS>
    </GET_GEOCODE>
  </REQUEST>
</ARCXML>
```

ARCXML

Used in: CONFIG, REQUEST, RESPONSE
Parent element: 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"?>

UTF-8 encoding is used with ArcXML 1.1. UTF-8 and UTF8 can be used interchangeably in map configuration files, viewer configuration files, requests, and responses. Responses from ArcIMS Route Server are returned with encoding=“UTF-8".

Attribute descriptions for ARCXML:

Attribute	Usage
version	Specifies version of ArcXML statement.

Example for ARCXML:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_ROUTE name="myroute">
      <STOPSLIST>
        <ROUTE_STOP id="0" desc="75080">
          <POINT x="-96.743072" y="32.971216"/>
        </ROUTE_STOP>
        <ROUTE_STOP id="1" desc="75081">
          <POINT x="-96.710713" y="32.948611"/>
        </ROUTE_STOP>
      </STOPSLIST>
      <ROUTEFILTER>
        <ENVELOPE minx="-180.0063613231552" miny="11.358778625954183" maxx="-64.99363867684477"
maxy="78.64122137404581"/>
      </ROUTEFILTER>
      <ROUTE_PROPERTIES>
        <ROUTETYPE>quickest</ROUTETYPE>
        <HWYPREF>50</HWYPREF>
        <PRECISION>0.1463266191428886</PRECISION>
        <LOCALE language="en" country="US" variant="" />
      <LENGTH_UNITS name="kilometer"/>
      </ROUTE_PROPERTIES>
    </GET_ROUTE>
  </REQUEST>
</ARCXML>
```

BARRIERSLIST

Used in: REQUEST
Parent element: GET_ROUTE

<BARRIERSLIST >

No attributes

(m) <BARRIER_POINT... />

</ BARRIERSLIST >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Defines a list of routing barriers. A barrier is a location on the road through which it is impossible to route at the present time. Barriers represent, for example, repair work, car accidents and traffic congestion.

Restrictions:

None

Notes:

Each barrier is snapped to the closest street segment, which makes it non-routable for the duration of a route calculation.

Example for BARRIERLIST:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ARCXML SYSTEM "request.dtd">
<ARCXML version="1.1">
 <REQUEST>
```

```
 <GET_ROUTE name="RouteTest for Barriers">
    <STOPSLIST>
       ...
    </STOPSLIST>
  <BARRIERSLIST>
    <BARRIER_POINT>
      <POINT x="-90.212855" y="38.611982"/>
    </BARRIER_POINT>
    <BARRIER_POINT>
      <POINT x="-89.908691" y="38.744964"/>
    </BARRIER_POINT>
  </BARRIERSLIST>
  <ROUTEFILTER>
   <ENVELOPE minx="-180.0063613231552" miny="11.358778625954183" maxx="-64.99363867684477"
maxy="78.64122137404581"/>
  </ROUTEFILTER>
   <ROUTE_PROPERTIES>
      <ROUTETYPE>shortest</ROUTETYPE>
      <HWYPREF>50</HWYPREF>
     <PRECISION>0.1463266191428886</PRECISION>
      <LOCALE language="en" country="US" variant="" />
      <LENGTH_UNITS name="kilometer"/>
  </ROUTE_PROPERTIES>
 </GET_ROUTE>
</REQUEST>
</ARCXML>
```

BARRIER_POINT

Used in: REQUEST
Parent element: BARRIERLIST

<BARRIER_POINT >

No attributes

<POINT... />

</ BARRIER POINT >

Bold: Attribute or child element is required.

Description:

Point representing a barrier.

Restrictions:

None

Notes:

Point coordinates can be in any projection. FILTERCOORDSYS must be set appropriately.

Example for BARRIER_POINT:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ARCXML SYSTEM "request.dtd">
<ARCXML version="1.1">
 <REQUEST>
```

```
  <GET_ROUTE name="RouteTest for Barriers">
    <STOPSLIST>
       ...
    </STOPSLIST>
  <BARRIERSLIST>
    <BARRIER_POINT>
      <POINT x="-90.212855" y="38.611982"/>
    </BARRIER_POINT>
    <BARRIER_POINT>
      <POINT x="-89.908691" y="38.744964"/>
    </BARRIER_POINT>
  </BARRIERSLIST>
  <ROUTEFILTER>
   <ENVELOPE minx="-180.0063613231552" miny="11.358778625954183" maxx="-64.99363867684477"
maxy="78.64122137404581"/>
  </ROUTEFILTER>
  <ROUTE_PROPERTIES>
      <ROUTETYPE>shortest</ROUTETYPE>
      <HWYPREF>50</HWYPREF>
     <PRECISION>0.1463266191428886</PRECISION>
      <LOCALE language="en" country="US" variant="" />
      <LENGTH_UNITS name="kilometer"/>
  </ROUTE_PROPERTIES>
  </GET_ROUTE>
 </REQUEST>
</ARCXML>
```

CLEARRESTRICTION

Used in: REQUEST
Parent element: RESTRICTIONS

<CLEARRESTRICTION
name =”*string*”
/>

Bold: Attribute or child element is required.

Description:

Turns off a restriction so it will not affect route calculation.

Notes:

Attribute “name” should specify an existent dataset restriction; otherwise, RouteServer responds with an error. Use GET_SERVICE_INFO to get a list of available restrictions.

Attribute descriptions for CLEARRESTRICTION:

Attribute	Usage
name	Name of the restriction as it is specified in the dataset (see SERVICEINFO/RESTRICTIOLIST)

Example for CLEARRESTRICTION:

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

```
  <REQUEST>
    <GET_ROUTE name=”RT_002">
       <STOPSLIST>
        <ROUTE_STOP id=”0" desc=”Lat: 38.47  Long:-121.51">
          <POINT x=”-121.51" y=”38.47"/>
         </ROUTE_STOP>
      <ROUTE_STOP id=”1" desc=”Lat: 38.49011585133329 Long: -121.45621949173282">
        <POINT x=”-121.45621949173282" y=”38.49011585133329"/>
        </ROUTE_STOP>
      <ROUTE_STOP id=”2" desc=”Lat: 38.48740970639922 Long: -121.45949914879973">
        <POINT x=”-121.45949914879973" y=”38.48740970639922"/>
        </ROUTE_STOP>
       </STOPSLIST>
   <ROUTE_PROPERTIES>
     <RESTRICTIONS>
      <SETRESTRICTION name=”Height” parameter = “5” type=”strict” />
      <CLEARRESTRICTION name=”CarPool” />
     </RESTRICTIONS>
   <ROUTETYPE>quickest</ROUTETYPE>
   <PRECISION>0.0005</PRECISION>
   </ROUTE_PROPERTIES>
  </GET_ROUTE>
 </REQUEST>
</ARCXML>
```

COORDS

Used in: RESPONSE
Parent element: PATH

<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- and 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 (;).

Example for COORDS:

```
<ARCXML version="1.1">
  <RESPONSE>
    <ROUTE name="myroute">
      <ENVELOPE minx="-118.366388" miny="34.164836" maxx="-118.332836"
maxy= "34.175727"/>
  <PARTS>
```

```
  <PART fromstop="0" tostop="1" >
   <SEGMENT distance="0.000000" time="0.000000">
    <ENVELOPE minx="-118.366388" miny="34.175727" maxx="-118.366388" maxy="34.175727"/>
    <DESCRIPTION type="depart">Depart Lat: 34.17556482872054
Long:-118.36638757243382</DESCRIPTION>
     <POLYLINE>
      <PATH>
         <COORDS>-118.366388 34.175727;-118.366388 34.175727</COORDS>
       </PATH>
     </POLYLINE>
    </SEGMENT>
   </PART>
 </PARTS>
 <TOTALS distance="4.000668" time="8.433333">
   <DESCRIPTION type="totals">Driving distance: 4 mile(s)</DESCRIPTION>
   <DESCRIPTION type="totals">Driving time: 8 minute(s)</DESCRIPTION>
 </TOTALS>
</ROUTE>
</RESPONSE>
</ARCXML>
```

DESCRIPTION

Used in: RESPONSE
Parent elements: SEGMENT, TOTALS

<DESCRIPTION

type =“*depart* | *daystart* | *dayend* | *arrive* | *direction* | *drivetext* | *totals*”

>

No child elements

</DESCRIPTION >

Bold: Attribute or child element is required

Description:

When parent element is SEGMENT, contains the textual description of the driving directions for a segment. When parent element is TOTALS, contains the footer text and total driving distance and time.

Restrictions:

None

Notes:

- Descriptions of types of “daystart” and “dayend” are generated only when DRIVING_WINDOW is set and a route takes more than one day.
- Each description represents a single line in the output. More than one line can be used for a description.

<DESCRIPTION type="depart">Depart Lat: 34.17556482872054 Long:-118.36638757243382**</DESCRIPTION**>

<DESCRIPTION type="daystart">Day 1**</DESCRIPTION>**

<DESCRIPTION type="dayend">End of day**</DESCRIPTION>**

<DESCRIPTION type="arrive">Arrive at Lat: 34.173099868 Long:-118.347498224**</DESCRIPTION>**

<DESCRIPTION type="direction">Turn left on N Floyd Rd **</DESCRIPTION>**

<DESCRIPTION type="drivetext">Drive 1.6 mile(s) ~ 3 minute(s) **</DESCRIPTION>**

<DESCRIPTION type="totals">Driving time: 8 minute(s)**</DESCRIPTION>**

Attribute descriptions for DESCRIPTION:

Attribute	Usage
type	Type of the driving direction string.
depart	Departure from the start or stop point, for example ‘Depart Fresno, CA’.
dayend	End of a day. Not reported in case of intraday routing.
daystart	Starting of a day. Not reported in case of intraday routing.
arrive	Arriving at the stop/finish point.
direction	Driving instruction (where to go next, where to turn, etc.).
drivetext	Driving distance both length and time, for example, ‘drive 14 mile(s)’.
totals	Trip summary description: total time, total length, total driving time.

Example for DESCRIPTION:

```
<?xml version=”1.0" encoding=”UTF-8"?>
<ARCXML version=”1.1">
 <RESPONSE>
  <ROUTE name=””>
   <ENVELOPE minx=”-122.494255" miny=”38.206298" maxx=”-73.427217" maxy=”45.474996"/>
 <PARTS>
```

```
<PART fromstop="0" tostop="1" >
 <SEGMENT distance="0.000000" time="0.000000" eta="12:00">
  <ENVELOPE minx="-73.955177" miny="44.518557" maxx="-73.955177" maxy="44.518557"/>
      <DESCRIPTION type="depart">Depart Start</DESCRIPTION>
    <POLYLINE>
    <PATH><COORDS>-73.955177 44.518557;-73.955177 44.518557</COORDS></PATH>
    </POLYLINE>
  </SEGMENT>
  <SEGMENT distance="0.000000" time="0.000000" eta="12:00">
   <ENVELOPE minx="-73.955177" miny="44.518557" maxx="-73.955177" maxy="44.518557"/>
        <DESCRIPTION type="daystart">Day 1</DESCRIPTION>
      <POLYLINE>
        <PATH><COORDS>-73.955177 44.518557;-73.955177 44.518557</COORDS></PATH>
      </POLYLINE>
   </SEGMENT>
  <SEGMENT distance="1.745241" time="4.188485" eta="12:00">
   <ENVELOPE minx="-73.972379" miny="44.518557" maxx="-73.955177" maxy="44.535043"/>
      <DESCRIPTION type="direction">Go North West on Alderbrook Rd toward Alder Brook Park Rd</
DESCRIPTION>
        <DESCRIPTION type="drivetext">Drive 1.7 mile(s) ~ 4 minute(s)</DESCRIPTION>
      <POLYLINE>...</POLYLINE>
   </SEGMENT>
   ...
  <SEGMENT distance="0.000000" time="0.000000" eta="18:00">
   <ENVELOPE minx="-77.129822" miny="44.235471" maxx="-77.129822" maxy="44.235471"/>
        <DESCRIPTION type="dayend">End of day</DESCRIPTION>
      <POLYLINE>...</POLYLINE>
   </SEGMENT>
  <SEGMENT distance="0.000000" time="0.000000" eta="09:00">
   <ENVELOPE minx="-77.129822" miny="44.235471" maxx="-77.129822" maxy="44.235471"/>
        <DESCRIPTION type="daystart">Day 2</DESCRIPTION>
      <POLYLINE>
        <PATH><COORDS>-77.129822 44.235471;-77.129822 44.235471</COORDS></PATH>
      </POLYLINE>
   </SEGMENT>
  <SEGMENT distance="236.508330" time="258.009084" eta="09:00">
   <ENVELOPE minx="-81.230418" miny="42.907452" maxx="-77.129822" maxy="44.235471"/>
```

```
        <DESCRIPTION type="direction">Stay on HWY 401 (Macdonald Cartier Fwy)</DESCRIPTION>
        <DESCRIPTION type="drivetext">Drive 236.5 mile(s) ~ 4 hour(s) 18 minute(s)</DESCRIPTION>
       ...
    </SEGMENT>
   <SEGMENT distance="0.049106" time="0.117876" eta="14:25">
     <ENVELOPE minx="-122.494255" miny="38.521606" maxx="-122.493699" maxy="38.522167"/>
        <DESCRIPTION type="direction">Turn right on York Ln</DESCRIPTION>
        <DESCRIPTION type="drivetext">Drive &lt; 0.1 mile(s) ~ &lt; 1 minute</DESCRIPTION>
        <POLYLINE>
         <PATH><COORDS>-122.494255 38.521606;-122.493699 38.522167</COORDS></PATH>
        </POLYLINE>
    </SEGMENT>
   <SEGMENT distance="0.000000" time="0.000000" eta="14:25">
     <ENVELOPE minx="-122.493699" miny="38.522167" maxx="-122.493699" maxy="38.522167"/>
        <DESCRIPTION type="arrive">Arrive at Finish, on the right</DESCRIPTION>
        <POLYLINE>
         <PATH><COORDS>-122.493699 38.522167;-122.493699 38.522167</COORDS></PATH>
        </POLYLINE>
    </SEGMENT>
   </PART>
   </PARTS>
  <TOTALS distance="3051.718803" time="3385.214602">
     <DESCRIPTION type="totals">Driving distance: 3051.7 mile(s)</DESCRIPTION>
    <DESCRIPTION type="totals">Total travel time: 6 day(s) 5 hour(s) 25 minute(s)</DESCRIPTION>
    <DESCRIPTION type="totals">Driving time: 56 hour(s) 25 minute(s)</DESCRIPTION>
  </TOTALS>
 </ROUTE>
</RESPONSE>
</ARCXML>
```

DRIVETIME

Used in: RESPONSE
Parent element: RESPONSE

<DRIVETIME >

(m)<DTZONERESULT.../>

<ENVELOPE.../>

</DRIVETIME >

Bold: Attribute or child element is required

(m): Child element can be used multiple times

Description:

Contains information about resulting drive time zones

Restrictions:

None

Notes:

See GET_DRIVETIME for request

Number of zones found is equal to the number of zones requested in a GET_DRIVETIME request

Example for DRIVETIME:

```
<?xml version=”1.0" encoding=”UTF-8"?>
<ARCXML version=”1.1">
<RESPONSE>
   <DRIVETIME>
   <DTZONERESULT id=”1" limit=”0.100000">
     <POLYGON coords=”-82.940256 33.113230;-82.939382 33.113473;-82.940256 33.113230;-82.940741
33.113080;-82.941047 33.112975;-82.941269 33.112887;-82.941097 33.112960;-82.940546 33.113149;-
82.940256 33.113230" />
   </DTZONERESULT>
    <ENVELOPE minx=”-82.941269" miny=”33.112887" maxx=”-82.939382" maxy=”33.113473"/>
     </DRIVETIME>
</RESPONSE>
</ARCXML>
```

DRIVING_DAY

Used in: REQUEST
Parent element: ROUTE_PROPERTIES

<DRIVING_DAY >

No attributes

<DRIVING_WINDOW.../>

</DRIVING_DAY >

Description:

Used to specify trip related information.

Example for DRIVING_DAY:

```
<?xml version="1.0" encoding="UTF-8" ?><ARCXML version="1.1">
<REQUEST>
<GET_ROUTE name="1">
 <STOPSLIST>
  <ROUTE_STOP id="1" desc="Start"><POINT x="-71.47288298" y="41.786606252"/></ROUTE_STOP>
  <ROUTE_STOP id="1" desc="Stop1"><POINT x="-71.43191514" y="41.778305577"/></ROUTE_STOP>
  <ROUTE_STOP id="2" desc="Finish"><POINT x="-71.45869150" y="41.742425244"/></ROUTE_STOP>
 </STOPSLIST>
 <ROUTE_PROPERTIES>
  <TRIP_START>2006-09-26 17:55</TRIP_START>
  <DRIVING_DAY>
   <DRIVING_WINDOW>
    <START_TIME>09:00</START_TIME>
    <END_TIME>18:00</END_TIME>
```

```
   </DRIVING_WINDOW>
  </DRIVING_DAY>
  <ROUTETYPE>quickest</ROUTETYPE> <HWYPREF>50</HWYPREF> <PRECISION>0.00005</PRECISION>
 </ROUTE_PROPERTIES>
</GET_ROUTE>
</REQUEST>
</ARCXML>
```

DRIVING_WINDOW

Used in: REQUEST
Parent element: DRIVING_DAY

<DRIVING_WINDOW >

No attributes

(r)<START_TIME.../>

(r)<END_TIME.../>

</DRIVING_WINDOW >

(r): Attribute or child element is required.

Description:

Defines a driving window when a driver plans to drive. Resulting route will be split by days with appropriate directions.

Notes:

- TRIP_START should be set when DRIVING_WINDOW is used. Otherwise an unexpected result may be received. When DRIVING_WINDOW is specified but TRIP_START is not the RouteServer generates an implicit value of TRIP_START which is a combination of current date and START_TIME. Such scenario should be avoided.
- DIRECTIONS elements of types “daystart” and “dayend” will be reported in case of multiday route if DRIVING_WINDOW is set.
- Total travel time DIRECTION item is always reported when DRIVING_WINDOW is set.

Example for DRIVING_WINDOW:

```
<?xml version=”1.0" encoding=”UTF-8" ?><ARCXML version=”1.1">
<REQUEST>
<GET_ROUTE name=”1">
 <STOPSLIST>
  <ROUTE_STOP id=”1" desc=”Start”><POINT x=”-71.47288298" y=”41.786606252"/></ROUTE_STOP>
  <ROUTE_STOP id=”1" desc=”Stop1"><POINT x=”-71.43191514" y=”41.778305577"/></ROUTE_STOP>
  <ROUTE_STOP id=”2" desc=”Finish”><POINT x=”-71.45869150" y=”41.742425244"/></ROUTE_STOP>
 </STOPSLIST>
 <ROUTE_PROPERTIES>
  <TRIP_START>2006-09-26 17:55</TRIP_START>
  <DRIVING_DAY>
   <DRIVING_WINDOW>
    <START_TIME>09:00</START_TIME>
    <END_TIME>18:00</END_TIME>
   </DRIVING_WINDOW>
  </DRIVING_DAY>
```

DTCENTER

Used in: REQUEST
Parent element: GET_DRIVETIME

```
<DTCENTER >
   <POINT.../>
</DTCENTER >
```

Bold: Attribute or child element is required

Description:

Defines a center point for drive time zones search

Restrictions:

None

Notes:

Drive time zones search direction (to of from a center point) is defined by the DTPROPERTIES/DTDIRECTION element of the GET_DRIVETIME request.

Example for DTCENTER:

```
<?xml version=”1.0" encoding=”UTF-8" ?>
<ARCXML version=”1.1">
<REQUEST>
  <GET_DRIVETIME>
   <DTPROPERTIES>
     <DTTYPE>quickest</DTTYPE>
     <SPEED_FACTOR>0.861</SPEED_FACTOR>
     <DTDIRECTION>inbound</DTDIRECTION>
   </DTPROPERTIES>
     <DTCENTER>
     <POINT x=”-80.21351318359315" y=”25.7802246093748"/>
     </DTCENTER>
   <DTZONES>
     <DTZONEINPUT limit=”15" />
     <DTZONEINPUT limit=”30" />
     <DTZONEINPUT limit=”60" />
   </DTZONES>
   <FEATURECOORDSYS id=”54030" />
  </GET_DRIVETIME>
</REQUEST>
</ARCXML>
```

DTDIRECTION

Used in: REQUEST
Parent element: DTPROPERTIES

<DTDIRECTION >

No attribute

No child elements

</DTDIRECTION >

Description:

This element defines direction of the drive time zones to be found relatively to the center point.

Restrictions:

The valid content words are inbound and outbound.

<DTDIRECTION>inbound**</DTDIRECTION>**

<DTDIRECTION>outbound**</DTDIRECTION>**

Notes:

Outbound direction means to search a zone when each point of this zone is accessible **from the center location**. Inbound search results to finding a zone when center location is accessible **from any point of the zone**.

Example for DTDIRECTION:

```
<?xml version=”1.0" encoding=”UTF-8" ?>
<ARCXML version=”1.1">
<REQUEST>
  <GET_DRIVETIME>
    <DTPROPERTIES>
      <DTTYPE>quickest</DTTYPE>
      <SPEED_FACTOR>0.861</SPEED_FACTOR>
       <DTDIRECTION>inbound</DTDIRECTION>
    </DTPROPERTIES>
    <DTCENTER>
      <POINT x=”-80.21351318359315" y=”25.7802246093748"/>
    </DTCENTER>
    <DTZONES>
      <DTZONEINPUT limit=”15" />
      <DTZONEINPUT limit=”30" />
      <DTZONEINPUT limit=”60" />
    </DTZONES>
    <FEATURECOORDSYS id=”54030" />
  </GET_DRIVETIME>
</REQUEST>
</ARCXML>
```

DTPROPERTIES

Used in: GET_DRIVETIME
Parent element: DTPROPERTIES

<DTPROPERTIES >

No attribute

<DTTYPE... />

<SPEED_FACTOR... />
<LOCALE…/>
<DTDIRECTION…/>

</DTPROPERTIES >

Bold: Attribute or child element is required

Description:

It is a parent element for drive time zones search properties.

Restrictions:

None

Notes:

- If the SPEED_FACTOR child element is omitted then speed factor defaults to 1 (no speed editing).
- If the DTDIRECTION child element is omitted then outbound drive time zones are searched by default.
- If the LOCALE element is omitted, the locale defined in the map service will be used.

Example for DTPROPERTIES:

```
<?xml version=”1.0" encoding=”UTF-8" ?>
<ARCXML version=”1.1">
<REQUEST>
  <GET_DRIVETIME>
     <DTPROPERTIES>
       <DTTYPE>quickest</DTTYPE>
       <SPEED_FACTOR>0.861</SPEED_FACTOR>
       <DTDIRECTION>inbound</DTDIRECTION>
     </DTPROPERTIES>
     <DTCENTER>
       <POINT x=”-80.21351318359315" y=”25.7802246093748"/>
     </DTCENTER>
     <DTZONES>
       <DTZONEINPUT limit=”15" />
       <DTZONEINPUT limit=”30" />
       <DTZONEINPUT limit=”60" />
     </DTZONES>
     <FEATURECOORDSYS id=”54030" />
  </GET_DRIVETIME>
</REQUEST>
</ARCXML>
```

DTTYPE

Used in: REQUEST
Parent element: DTPROPERTIES

<DTTYPE >

No attribute

No child elements

</DTTYPE >

Description:

This element defines type of the drive time zone limit.

Restrictions:

The valid content words are quickest and shortest.

<DTTYPE>quickest**</DTTYPE>**

<DTTYPE>shortest**</DTTYPE>**

Notes:

Zone sizes are specified in minutes when DTTYPE is quickest and in miles when shortest.

Example for DTTYPE:

```
<?xml version=”1.0" encoding=”UTF-8" ?>
<ARCXML version=”1.1">
<REQUEST>
  <GET_DRIVETIME>
    <DTPROPERTIES>
      <DTTYPE>quickest</DTTYPE>
      <SPEED_FACTOR>0.861</SPEED_FACTOR>
      <DTDIRECTION>inbound</DTDIRECTION>
    </DTPROPERTIES>
    <DTCENTER>
      <POINT x=”-80.21351318359315" y=”25.7802246093748"/>
    </DTCENTER>
    <DTZONES>
      <DTZONEINPUT limit=”15" />
      <DTZONEINPUT limit=”30" />
      <DTZONEINPUT limit=”60" />
    </DTZONES>
    <FEATURECOORDSYS id=”54030" />
  </GET_DRIVETIME>
</REQUEST>
</ARCXML>
```

DTZONEINPUT

Used in: REQUEST
Parent element: DTZONES

```
<DTZONEINPUT
   limit ="double"
   >
</DTZONEINPUT >
```

Bold: Attribute or child element is required

(m): Child element can be used multiple times

Description:

The element specifies size of the zone, which has to be found while processing GET_DRIVETIME request.

Restrictions:

None

Notes:

Drive time zone limit's unit (minutes or mile) is defined by the DTPROPERTIES/DTTYPE element of the request.

Example for DTZONEINPUT:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
<REQUEST>
  <GET_DRIVETIME>
    <DTPROPERTIES>
        <DTTYPE>quickest</DTTYPE>
      <SPEED_FACTOR>0.861</SPEED_FACTOR>
      <DTDIRECTION>inbound</DTDIRECTION>
    </DTPROPERTIES>
    <DTCENTER>
      <POINT x="-80.21351318359315" y="25.7802246093748"/>
    </DTCENTER>
    <DTZONES>
         <DTZONEINPUT limit="15" />
         <DTZONEINPUT limit="30" />
         <DTZONEINPUT limit="60" />
    </DTZONES>
    <FEATURECOORDSYS id="54030" />
  </GET_DRIVETIME>
</REQUEST>
</ARCXML>
```

DTZONERESULT

Used in: REQUEST
Parent element: DRIVETIME

```
<DTZONERESULT >
   limit ="double"
   >
<POLYGON ... />
</DTZONERESULT >
```

Bold: Attribute or child element is required

Description:

This element contains coordinates of the found driving time zone's polygon.

Restrictions:

None

Notes:

None

Example for DTZONERESULT:

```
<?xml version=”1.0" encoding=”UTF-8"?>
<ARCXML version=”1.1">
<RESPONSE>
  <DRIVETIME>
      <DTZONERESULT  limit=”1">
      <POLYGON coords=”-82.940256 33.113230;-82.939382 33.113473;-82.940256 33.113230;-82.940741
33.113080;-82.941047 33.112975;-82.941269 33.112887;-82.941097 33.112960;-82.940546 33.113149;-
82.940256 33.113230" />
      </DTZONERESULT>
    <ENVELOPE minx=”-82.941269" miny=”33.112887" maxx=”-82.939382" maxy=”33.113473"/>
  </DRIVETIME>
</RESPONSE>
</ARCXML>
```

DTZONES

Used in: REQUEST
Parent element: GET_DRIVETIME

<DTZONES >

(m) <DTZONEINPUT... />

</DTZONES >

Bold: Attribute or child element is required

(m): Child element can be used multiple times

Description:

The element specifies a driving time zones list.

Restrictions:

At least one zone has to be set.

Notes:

None

Example for DTZONES:

```
<?xml version=”1.0" encoding=”UTF-8" ?>
<ARCXML version=”1.1">
<REQUEST>
  <GET_DRIVETIME>
    <DTPROPERTIES>
      <DTTYPE>quickest</DTTYPE>
      <SPEED_FACTOR>0.861</SPEED_FACTOR>
      <DTDIRECTION>inbound</DTDIRECTION>
    </DTPROPERTIES>
    <DTCENTER>
      <POINT x=”-80.21351318359315" y=”25.7802246093748"/>
    </DTCENTER>
      <DTZONES>
      <DTZONEINPUT limit=”15" />
      <DTZONEINPUT limit=”30" />
      <DTZONEINPUT limit=”60" />
      </DTZONES>
     <FEATURECOORDSYS id=”54030" />
  </GET_DRIVETIME>
</REQUEST>
</ARCXML>
```

END_TIME

Used in: REQUEST
Parent element: DRIVING_WINDOW

<END_TIME >

No attributes

No child elements

</END_TIME >

Description:

Describes a time of day when current day driving should finish. Used for multiday routes.

Restrictions:

- The value of the element should be specified in the format hh:mm; otherwise, an error will be returned.
- END_TIME should be later than START_TIME.

Example for END_TIME:

```
<?xml version=”1.0" encoding=”UTF-8" ?><ARCXML version=”1.1">
<REQUEST>
<GET_ROUTE name=”1">
 <STOPSLIST>
  <ROUTE_STOP id=”1" desc=”Start”><POINT x=”-71.47288298" y=”41.786606252"/></ROUTE_STOP>
  <ROUTE_STOP id=”1" desc=”Stop1"><POINT x=”-71.43191514" y=”41.778305577"/></ROUTE_STOP>
  <ROUTE_STOP id=”2" desc=”Finish”><POINT x=”-71.45869150" y=”41.742425244"/></ROUTE_STOP>
 </STOPSLIST>
```

```
 <ROUTE_PROPERTIES>
  <TRIP_START>2006-09-26 17:55</TRIP_START>
  <DRIVING_DAY>
   <DRIVING_WINDOW>
    <START_TIME>09:00</START_TIME>
    <END_TIME>18:00</END_TIME>
   </DRIVING_WINDOW>
  </DRIVING_DAY>
  <ROUTETYPE>quickest</ROUTETYPE> <HWYPREF>50</HWYPREF> <PRECISION>0.00005</PRECISION>
 </ROUTE_PROPERTIES>
</GET_ROUTE>
</REQUEST>
</ARCXML>
```

ENVELOPE

Used in: REQUEST, RESPONSE
Parent elements: ROUTE, ROUTEFILTER, SEGMENT, DRIVETIME

```
<ENVELOPE
    maxx =“double”
    maxy =“double”
    minx =“double”
    miny =“double”
>
    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.

Restrictions:

None

Notes:

None

Attribute descriptions for ENVELOPE:

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.

Examples for ENVELOPE:

Example 1: When in **ROUTEFILTER** in a **GET_ROUTE** request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_ROUTE name="myroute">
      <STOPSLIST>
        <ROUTE_STOP id="0" desc="75080">
          <POINT x="-96.743072" y="32.971216"/>
        </ROUTE_STOP>
        <ROUTE_STOP id="1" desc="75081">
          <POINT x="-96.710713" y="32.948611"/>
        </ROUTE_STOP>
      </STOPSLIST>
      <ROUTEFILTER>
        <ENVELOPE minx="-180.0063613231552" miny="11.358778625954183" maxx="-64.99363867684477"
maxy="78.64122137404581"/>
      </ROUTEFILTER>
      <ROUTE_PROPERTIES>
        <ROUTETYPE>quickest</ROUTETYPE>
        <HWYPREF>50</HWYPREF>
        <PRECISION>0.1463266191428886</PRECISION>
         <LOCALE language="en" country="US" variant="" />
       <LENGTH_UNITS name="kilometer"/>
      </ROUTE_PROPERTIES>
```

```
    </GET_ROUTE>
  </REQUEST>
</ARCXML>
```

Example 2: When in a **ROUTE** response.

```
<ARCXML version="1.1">
<RESPONSE>
<ROUTE name="">
  <ENVELOPE minx="-118.366388" miny="34.164836" maxx="-118.332836"
maxy="34.175727"/>
  <PARTS>
   <PART fromstop="0" tostop="1" >
    <SEGMENT distance="0.000000" time="0.000000">
     <ENVELOPE minx="-118.366388" miny="34.175727" maxx=”-118.366388" maxy="34.175727"/>
     <DESCRIPTION type="depart">Depart Lat: 34.17556482872054
Long:-118.36638757243382</DESCRIPTION>
     <POLYLINE>
      <PATH>
        <COORDS>-118.366388 34.175727;-118.366388 34.175727</COORDS>
       </PATH>
     </POLYLINE>
    </SEGMENT>
   </PART>
 </PARTS>
 <TOTALS distance="4.000668" time="8.433333">
    <DESCRIPTION type="totals">Driving distance: 4 mile(s)</DESCRIPTION>
    <DESCRIPTION type="totals">Driving time: 8 minute(s)</DESCRIPTION>
 </TOTALS>
</ROUTE>
</RESPONSE>
</ARCXML>
```

Example 3: When in a **DRIVETIME** response.

```
<?xml version=”1.0" encoding=”UTF-8"?>
<ARCXML version=”1.1">
<RESPONSE>
  <DRIVETIME>
    <DTZONERESULT id=”1" limit=”0.100000">
      <POLYGON coords=”-82.940256 33.113230;-82.939382 33.113473;-82.940256 33.113230;-82.940741
33.113080;-82.941047 33.112975;-82.941269 33.112887;-82.941097 33.112960;-82.940546 33.113149;-
82.940256 33.113230" />
    </DTZONERESULT>
    <ENVELOPE minx=”-82.941269" miny=”33.112887" maxx=”-82.939382" maxy=”33.113473"/>
  </DRIVETIME>
</RESPONSE>
</ARCXML>
```

ERROR

Used in: RESPONSE
Parent element: ROUTE

<ERROR>
machine =“*string*”
processid =“*integer*”
threadid =“*integer*”

No child elements

</ERROR >

Bold: Attribute or child element is required.

Description:

Gives an error message from the ArcIMS Spatial Server when attribute information in an element does not parse correctly or an error with the Route Server has occurred.

Restrictions:

None

Notes:

The following errors are returned for Route Server:

- For general errors:

<ERROR>Routing error occurred. Check server configuration.**</ERROR>**
This error occurs if Route Server files become corrupt or an out of memory situation occurs.

- In response to GET_ROUTE
  - When a route cannot be found, for example, when one of the stops is located on an island with no connecting road (in the case of Hawaii and the continental U.S.).

    <ERROR>Cannot find route between specified stops. Probably no road exists connecting the stops. **</ERROR>**

  - When no street segment can be found in proximity from stop location:

    <ERROR>Stop cannot be placed on routable road. Stop id="<stop_id>". **</ERROR>**

  - When the streets data does not have appropriate index for specified type of route, for example, if the data does not have traversal time attribute defined, it is impossible to calculate quickest routes:

    <ERROR>Selected route type is not supported.**</ERROR>**

  - When there are not enough stops to generate a route:

    <ERROR>At least two stops required to generate a route.**</ERROR>**

  - When a barrier is placed at a location that is not on a routable street.

    <ERROR>Stop cannot be placed on a routable road. **</ERROR>**

Attribute descriptions for ARCXML:

Attribute	Usage
machine	Machine name where the ArcIMS Spatial Server processing the request resides.
process_id	Unique identifier for the ArcIMS Spatial Server that processed the request.
threadid	ArcIMS Spatial Server instance (thread) processing the request.

Example for ERROR:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <ROUTE name="myroute”>
      <ERROR machine="RouteServer" processid="440" threadid="398">
        Selected route type is not supported.
      </ERROR>
    </ROUTE>
  </RESPONSE>
</ARCXML>
```

EXTENSION

Used in: RESPONSE
Parent element: LAYERINFO

```
<EXTENSION>
    type =“route | sdcgeocode” >
   <GCSTYLE…/> [Or]
<RESTRICTIONSLIST…/>
</EXTENSION >
```

Bold: Attribute or child element is required.

Description:

Defines routing and geocoding parameters for Spatial Data Compressed (SDC) data layers. In CONFIG, each EXTENSION tag defines one routing or geocoding service.

Restrictions:

None

Notes:

None

Attribute descriptions for EXTENSION:

Attribute	Usage
type	Extension type.

Example for EXTENSION:

Example1: When in **RESPONSE** from Route Server

```
<?xml version=”1.0" encoding=”UTF-8"?>
<ARCXML version=”1.1">
<RESPONSE>
  <SERVICEINFO>
    <LAYERINFO type=”featureclass” id=”Layer6" name=”Streets” >
      <FCLASS type=”line” />
        <EXTENSION  type=”Route”>
        <RESTRICTIONSLIST>
          <RESTRICTION name=”Carpool” type=”boolean” />
          <RESTRICTION name=”Pedestrian” type=”boolean” />
          <RESTRICTION name=”Height” type=”maxallowed” />
          <RESTRICTION name=”Weight” type=”maxallowed” />
        </RESTRICTIONSLIST>
       </EXTENSION>
    </LAYERINFO>
  </SERVICEINFO>
</RESPONSE>
</ARCXML>
```

Example 2: When in **CONFIG**

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
 <CONFIG>
  <MAP>
   <LAYER type="featureclass" name="Streets" visible="true" id="Streets" maxscale="1:45700">
     <DATASET name="streets" type="line" workspace="sws_sdc2"/>
        <EXTENSION  type="SDCGeocode">
       <GCSTYLE name="SDC.LINE_STREET_ADDRESS.USA">
         <GCFIELD id="L_F_ADD" name="L_F_ADD"/>
         <GCFIELD id="R_F_ADD" name="R_F_ADD"/>
         <GCFIELD id="L_T_ADD" name="L_T_ADD"/>
         <GCFIELD id="R_T_ADD" name="R_T_ADD"/>
         <GCFIELD id="PREFIX" name="PREFIX"/>
         <GCFIELD id="NAME" name="NAME"/>
         <GCFIELD id="TYPE" name="TYPE"/>
         <GCFIELD id="SUFFIX" name="SUFFIX"/>
            ...
         <GCPARAM id="Offset" value="10"/>
        </GCSTYLE>
       </EXTENSION>
   </LAYER>
  </MAP>
 </CONFIG>
</ARCXML>
```

Example3: When in **RESPONSE** from SDC Geocode Server

```
<?xml version=”1.0" encoding=”UTF-8"?>
<ARCXML version=”1.1">
<RESPONSE>
  <SERVICEINFO>
    <LAYERINFO type=”featureclass” name=”Street Address” id=”StreetAddress”>
      <FCLASS type=”line”/>
         <EXTENSION  type=”SDCGeocode”>
```

```
      <GCSTYLE name=”SDC.DISP_STREET_ADDRESS.NA” >
      <GCINPUT id=”Address or Intersection” label=”Address or Intersection” width=”255" type=”text”
description=”Address or Intersection” />
      <GCINPUT id=”ZIP” label=”ZIP or FSA Code” width=”0" type=”text” description=”ZIP or FSA Code” />
      <GCINPUT id=”City” label=”City” width=”255" type=”text” description=”City” />
      <GCINPUT id=”State” label=”State or Province” width=”2" type=”text” description=”State or Prov-
ince” />
       </GCSTYLE>
      </EXTENSION>
     </LAYERINFO>
  </SERVICEINFO>
</RESPONSE>
</ARCXML>
```

FEATURE

Used in: RESPONSE
Parent element: GEOCODE

<FEATURE>

featureid =“*integer*” >
(m) <FIELD... />
</FEATURE >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Defines framework for geocoding results.

Restrictions:

None

Notes:

None

Attribute descriptions for FEATURE:

Attribute	Usage
featureid	Provides a unique value for a feature. Each value is assigned sequentially.

Example for FEATURE:

```
<?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="34" precision="0">
         <FIELDVALUE valuestring="162 Golder Rd, 04240, ME, Lewiston" />
      </FIELD>
      <FIELD type="-98" name="SHAPEFIELD">
          <FIELDVALUE>
            <POINT x="-70.147315455731" y="44.102606073962" />
          </FIELDVALUE>
      </FIELD>
      <FIELD type="12" name="House Range” size="3" precision="0">
         <FIELDVALUE valuestring="162" />
      </FIELD>
      <FIELD type="12" name="Name” size="9" precision=”0">
         <FIELDVALUE valuestring="Golder Rd" />
      </FIELD>
      <FIELD type="12" name="Postal Code" size="5" precision="0">
         <FIELDVALUE valuestring="04240" />
      </FIELD>
      <FIELD type=”12" name="City” size="8" precision="0">
         <FIELDVALUE valuestring="Lewiston" />
      </FIELD>
      <FIELD type="12" name="State or Province" size="2" precision="0">
         <FIELDVALUE valuestring="ME" />
      </FIELD>
      <FIELD type="12" name="Country" size="2" precision="0">
         <FIELDVALUE valuestring="US" />
      </FIELD>
     </FEATURE>
    <GCCOUNT count="1" />
  </GEOCODE>
</RESPONSE>
</ARCXML>
```

FEATURECOORDSYS

Used in: REQUEST, CONFIG
Parent element: PROPERTIES, GET_ROUTE, GET_DRIVETIME, GET_GEOCODE

```
<FEATURECOORDSYS
    id =“integer” [Or]
   string =“string” [Or]
>
   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.
- Datum transformations are not supported.

Notes:

- See *ArcXML Programmer's Reference* in the ArcIMS help for a complete list of supported IDs and definition strings.
- 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 (all one line):

PROJCS[“World_Mollweide”,GEOGCS[“GCS_WGS_1984”,DATUM[“D_WGS_1984”,SPHEROID[“WGS_1984”,6378137,298.257223563]] ,PRIMEM[“Greenwich”,0],UNIT[“Degree”,0.017453292519943295]],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 (all one line):

PROJCS["World_Mollweide",GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID ["WGS_1984",6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",0.017453292519943295]],

PROJECTION["Mollweide"],PARAMETER["False_Easting",0],PARAMETER["False_Northing",0], PARAMETER["Central_Meridian",0],UNIT["Meter",1]]

Attribute descriptions for FEATURECOORDSYS:

Attribute	Usage
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.

Example for FEATURECOORDSYS:

Example: When in a **GET_ROUTE** request

```
<ARCXML version="1.1">
<REQUEST>
<GET_ROUTE name="Stops”>
  <STOPSLIST>
    <ROUTE_STOP id="Stop0" desc="Stop0">
      <POINT x="-1042477.6889912172" y="5100206.304991311"/>
    </ROUTE_STOP>
    <ROUTE_STOP id="Stop1" desc="Stop1">
      <POINT x="-1259958.7370845077" y="4792833.090352795"/>
    </ROUTE_STOP>
  </STOPSLIST>
  <ROUTE_PROPERTIES>
    <ROUTETYPE>quickest</ROUTETYPE>
    <HWYPREF>50</HWYPREF>
    <PRECISION>0.1463266191428886</PRECISION>
    <LOCALE language="en" country="US" variant="" />
    <LENGTH_UNITS name="kilometer"/>
  </ROUTE_PROPERTIES>
   <FEATURECOORDSYS id="21893"/>
   <FILTERCOORDSYS id="21893"/>
</GET_ROUTE>
</REQUEST>
</ARCXML>
```

FIELD

Used in: RESPONSE
Parent element: FEATURE

```
<FIELD
    name =“string”
     type =“-98 | 4 | 12"
    precision =“integer”
    size =“integer” >
   <FIELDVALUE... />
</FIELD >
```

Bold: Attribute or child element is required.

Description:

Contains the geocoded address information.

Restrictions:

None

Notes:

None

Attribute descriptions for FIELD:

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
-98	Shape fields.
4	Integer fields. Values can range between -2147483648 and 2147483647 or 0 and 4294967295.
12	String fields of any length.

Example for FIELD:

```
<?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="34" precision="0">
          <FIELDVALUE valuestring="162 Golder Rd, 04240, ME, Lewiston" />
        </FIELD>
       <FIELD type=”-98" name=”SHAPEFIELD”>
          <FIELDVALUE>
            <POINT x="-70.147315455731" y="44.102606073962" />
          </FIELDVALUE>
        </FIELD>
       <FIELD type="12" name="House Range" size="3" precision="0">
          <FIELDVALUE valuestring="162" />
        </FIELD>
       <FIELD type="12" name="Name" size="9" precision="0">
          <FIELDVALUE valuestring="Golder Rd" />
        </FIELD>
       <FIELD type="12" name="Postal Code" size="5" precision="0">
```

```
        <FIELDVALUE valuestring="04240" />
      </FIELD>
      <FIELD type="12" name="City" size="8" precision="0">
        <FIELDVALUE valuestring="Lewiston" />
      </FIELD>
      <FIELD type="12" name="State or Province" size="2" precision="0">
        <FIELDVALUE valuestring="ME" />
      </FIELD>
      <FIELD type="12" name="Country" size="2" precision="0">
        <FIELDVALUE valuestring="US" />
      </FIELD>
    </FEATURE>
    <GCCOUNT count="1" />
  </GEOCODE>
</RESPONSE>
</ARCXML>
```

FIELDVALUE

Used in: RESPONSE
Parent element: FIELD

```
<FIELDVALUE
   valuestring ="string" >
    <POINT... />
</FIELDVALUE >
```

Description:

Sets the value of a FIELD in a GEOCODE response.

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.

Example for FIELDVALUE:

```
<?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="34" precision="0">
          <FIELDVALUE valuestring="162 Golder Rd, 04240, ME, Lewiston" />
      </FIELD>
      <FIELD type="-98" name="SHAPEFIELD">
          <FIELDVALUE>
           <POINT x="-70.147315455731" y="44.102606073962" />
          </FIELDVALUE>
      </FIELD>
      <FIELD type="12" name="House Range" size="3" precision="0">
          <FIELDVALUE valuestring="162" />
      </FIELD>
      <FIELD type="12" name="Name" size="9" precision="0">
          <FIELDVALUE valuestring=”Golder Rd” />
      </FIELD>
      <FIELD type="12" name="Postal Code" size="5" precision="0">
          <FIELDVALUE valuestring="04240" />
      </FIELD>
      <FIELD type="12" name="City" size="8" precision="0">
          <FIELDVALUE valuestring=”Lewiston” />
      </FIELD>
      <FIELD type="12" name="State or Province" size="2" precision="0">
          <FIELDVALUE valuestring="ME" />
      </FIELD>
      <FIELD type="12" name="Country" size="2" precision="0">
          <FIELDVALUE valuestring=”US” />
      </FIELD>
    </FEATURE>
    <GCCOUNT count="1" />
  </GEOCODE>
</RESPONSE>
</ARCXML>
```

FILTERCOORDSYS

Used in: REQUEST, CONFIG
Parent elements: PROPERTIES, GET_ROUTE, GET_DRIVETIME

<FILTERCOORDSYS
id="*integer*" *[Or]*
string="*string*" *[Or]*
>
No child elements
</FILTERCOORDSYS >

Bold: Attribute or child element is required.

Description:

The current coordinate system of filters in requesting client.

Restrictions:

- Must use either *id* or *string*, but not both.
- Datum transformations are not supported.

Notes:

See *ArcXML Programmer's Reference* in the ArcIMS help for a complete list of supported IDs and definition strings.

- 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 (all one line):

PROJCS["World_Mollweide",GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137,298.257223563]], PRIMEM["Greenwich",0],UNIT["Degree",0.0174532925199432957]],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 (all one line):

PROJCS["World_Mollweide",GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",0.017453292519943295]],PROJECTION["Mollweide"],PARAMETER["False_Easting",0],PARAMETER["False_Northing",0],PARAMETER["Central_Meridian",0],UNIT["Meter",1]]

Attribute descriptions for FILTERCOORDSYS:

Attribute	Usage
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.

Example for FILTERCOORDSYS:

Example 1: When in a **GET_ROUTE** request

```
<ARCXML version="1.1">
<REQUEST>
<GET_ROUTE name="Stops">
  <STOPSLIST>
    <ROUTE_STOP id="Stop0" desc="Stop0">
      <POINT x="-1042477.6889912172" y="5100206.304991311"/>
    </ROUTE_STOP>
    <ROUTE_STOP id="Stop1" desc="Stop1">
      <POINT x="-1259958.7370845077" y="4792833.090352795"/>
    </ROUTE_STOP>
  </STOPSLIST>
  <ROUTE_PROPERTIES>
    <ROUTETYPE>quickest</ROUTETYPE>
    <HWYPREF>50</HWYPREF>
    <PRECISION>0.1463266191428886</PRECISION>
```

```
    <LOCALE language="en" country="US" variant="" />
    <LENGTH_UNITS name="kilometer"/>
  </ROUTE_PROPERTIES>
   <FILTERCOORDSYS id="21893"/>
</GET_ROUTE>
</REQUEST>
</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”/>
      </ENVIRONMENT>
      <MAP>
         <PROPERTIES>
            <MAPUNITS units="decimal_degrees"/>
           <ENVELOPE minx="-179.95" miny="18.34" maxx="-66.2" maxy="72.3" name="Initial_Extent"/>
            <FEATURECOORDSYS id="4269"/>
              <FILTERCOORDSYS id="4269"/>
            <BACKGROUND color="181,198,214"/>
         </PROPERTIES>
              ...
      </MAP>
   </CONFIG>
</ARCXML>
```

GCCOUNT

Used in: RESPONSE
Parent element: 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:

None

Attribute descriptions for GCCOUNT:

Attribute	Usage
count	Number of candidates found.

Example for GCCOUNT:

```
<?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="5" precision="0">
          <FIELDVALUE valuestring="89509" />
        </FIELD>
        <FIELD type="-98" name="SHAPEFIELD”>
          <FIELDVALUE>
            <POINT x="-119.90546116" y="39.45247355" />
          </FIELDVALUE>
        </FIELD>
      </FEATURE>
      <GCCOUNT count="1" />
    </GEOCODE>
  </RESPONSE>
</ARCXML>
```

GCINPUT

Used in: RESPONSE
Parent element: 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:

- All GCINPUT *id* attribute values are the same values used in GCTAG in a GET_GEOCODE request.
- GCINPUT for the streets layer contains information for both address matching and reverse geocoding.
    - Values for *id* depend on geocoding style. For example, for address matching, if the style is SDC.STREET_ADDRESS.USA, the values for *id* will be “Address_or_Intersection”, “ZIP”, “City”, and “State”. For reverse geocoding, if the style is LINE_STREET_REVERSE.USA, the values for *id* will be “X”, “Y”, “Style”, and “Direction”.

- GCINPUT attributes are not case sensitive.

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.

Example for GCINPUT:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
     <LAYERINFO type="featureclass" name="Street Address" id="StreetAddress" >
        <FCLASS type="line"/>
       <EXTENSION type="SDCGeocode">
         <GCSTYLE name="SDC.STREET_ADDRESS.USA" >
           <GCINPUT id="Address_or_Intersection" label="Address or Intersection" width="255"
type="text" description="Address or Intersection” />
           <GCINPUT id="ZIP" label="5-digit ZIP Code" width="5" type="text" description="5-digit ZIP
Code" />
           <GCINPUT id="City" label="City" width="255" type="text" description="City" />
           <GCINPUT id="State" label="State" width="2" type="text" description="State" />
          </GCSTYLE>
       </EXTENSION>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

GCPARAM

Used in: CONFIG
Parent element: GCSTYLE

< GCPARAM >
id =“*string*”
value =“*string*”
No child elements
</ GCPARAM >

Description:

Customizes geocoding style parameters. Please refer to *Geocoding Styles* for details on supported parameters.

Restrictions:

None

Notes:

None

Attribute Descriptions for GCPARAM:

Attribute	Usage
id	Specify the geocoder parameter.
value	Specify the geocoder parameter value.

Example for GCPARAM:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
 <CONFIG>
  <MAP>
   <LAYER type="featureclass" name="Streets" visible="true" id="Streets" maxscale="1:45700">
     <DATASET name="streets" type="line" workspace="sws_sdc2"/>
     <EXTENSION type="SDCGeocode">
       <GCSTYLE name="SDC.LINE_STREET_ADDRESS.USA">
         <GCFIELD id="L_F_ADD" name="L_F_ADD"/>
         <GCFIELD id="R_F_ADD" name="R_F_ADD"/>
         <GCFIELD id="L_T_ADD" name="L_T_ADD"/>
         <GCFIELD id="R_T_ADD" name="R_T_ADD"/>
         <GCFIELD id="PREFIX" name="PREFIX"/>
         <GCFIELD id="NAME" name="NAME"/>
         <GCFIELD id="TYPE" name="TYPE"/>
         <GCFIELD id="SUFFIX" name="SUFFIX"/>
            ...
          <GCPARAM id="Offset" value="10"/>
       </GCSTYLE>
     </EXTENSION>
   </LAYER>
  </MAP>
 </CONFIG>
</ARCXML>
```

GCSTYLE

Used in: RESPONSE
Parent element: EXTENSION

<GCSTYLE
name =“*string*”
>
*When parent element is **EXTENSION** in **CONFIG:***
(m) <GCFIELD... />

*When parent element is **EXTENSION** in **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:

None

Attribute descriptions for GCSTYLE:

Attribute	Usage
name	Geocoding style name. Please refer to *Geocoding Styles* for a complete list of geocoding styles supported by Route Server.

Example for GCSTYLE:

Example 1: When in a **SERVICEINFO** response.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
     <LAYERINFO type="featureclass" name="Street Address" id="StreetAddress" >
        <FCLASS type="line"/>
       <EXTENSION type="SDCGeocode">
          <GCSTYLE name="SDC.STREET_ADDRESS.USA" >
           <GCINPUT id="Address_or_Intersection" label="Address or Intersection" width="255"
type="text" description="Address or Intersection" />
           <GCINPUT id="ZIP” label="5-digit ZIP Code" width="5" type="text" description="5-digit ZIP
Code" />
           <GCINPUT id="City" label="City" width="255" type="text" description="City" />
           <GCINPUT id="State" label="State" width="2" type="text" description="State" />
           </GCSTYLE>
       </EXTENSION>
      </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

Example 2: When in **CONFIG**.

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

```
 <CONFIG>
  <MAP>
   <LAYER type=”featureclass” name=”Streets” visible=”true” id=”Streets” maxscale=”1:45700">
     <DATASET name=”streets” type=”line” workspace=”sws_sdc2"/>
     <EXTENSION type=”SDCGeocode”>
        <GCSTYLE name=”SDC.LINE_STREET_ADDRESS.USA”>
         <GCFIELD id=”L_F_ADD” name=”L_F_ADD”/>
         <GCFIELD id=”R_F_ADD” name=”R_F_ADD”/>
         <GCFIELD id=”L_T_ADD” name=”L_T_ADD”/>
         <GCFIELD id="R_T_ADD" name="R_T_ADD"/>
         <GCFIELD id="PREFIX" name="PREFIX"/>
         <GCFIELD id="NAME" name="NAME"/>
         <GCFIELD id="TYPE" name="TYPE"/>
         <GCFIELD id="SUFFIX" name="SUFFIX"/>
            ...
         <GCPARAM id="Offset" value="10"/>
        </GCSTYLE>
     </EXTENSION>
   </LAYER>
  </MAP>
 </CONFIG>
</ARCXML>
```

GCTAG

Used in: REQUEST
Parent element: 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. Reverse geocoding can also be used by supplying x, y coordinates.

Restrictions:

None

Notes:

- All GCTAG *id* attribute values need to match GCINPUT values returned in SERVICEINFO.
- For reverse geocoding, the following attributes are used:
  - When `id="x"` or `id="y"`, the *value* is the x- or y-coordinate in decimal degrees.
  - When `id="style"`, *value* can have three values. When `value="range"`, a numeric range of addresses is included in the response. When `value="address",` one address value is returned. When `value=“intersection”`, the nearest intersection is returned.
  - Only if the “`style`”=”intersection”, you may optionally specify “direction” attribute. When `id=”direction”`, the *value* is 0-360 degree azimuth.

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: For matching street addresses.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_GEOCODE maxcandidates="20" minscore="60">
      <LAYER id="18" />
        <ADDRESS>
          <GCTAG id="Street Address" value="380 new york St" />
          <GCTAG id="5-digit ZIP code" value="92373" />
          <GCTAG id="City" value="redlands" />
          <GCTAG id="State" value="ca" />
        </ADDRESS>
     </GET_GEOCODE>
  </REQUEST>
</ARCXML>
```

Example 2: For matching ZIP Codes.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_GEOCODE maxcandidates="20" minscore="60" >
      <LAYER id="24" />
        <ADDRESS>
          <GCTAG id="5-digit ZIP code" value="92373" />
        </ADDRESS>
     </GET_GEOCODE>
  </REQUEST>
```

```
</ARCXML>
```

Example 3: For reverse geocoding. When id="style", the *value* attribute can have a value of "range", "address" or "intersection".

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_GEOCODE maxcandidates="25" minscore="60">
      <LAYER id="18" />
      <ADDRESS>
        <GCTAG id="style" value="range" />
        <GCTAG id="x" value="-96.72772727" />
        <GCTAG id="y" value="32.96581818" />
      </ADDRESS>
    </GET_GEOCODE>
  </REQUEST>
</ARCXML>
```

Example 4: For reverse intersection geocoding. When style is intersection and direction is 90.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_GEOCODE maxcandidates="25" minscore="60">
      <LAYER id="18" />
      <ADDRESS>
        <GCTAG id="style" value="intersection" />
        <GCTAG id="direction" value="90" />
        <GCTAG id="x" value="-77.0132459997014" />
        <GCTAG id="y" value="38.8869969991112" />
      </ADDRESS>
    </GET_GEOCODE>
  </REQUEST>
</ARCXML>
```

GEOCODE

Used in: RESPONSE
Parent element: 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.

Example for GEOCODE:

```
<?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="34" precision="0">
             <FIELDVALUE valuestring="162 Golder Rd, 04240, ME, Lewiston" />
        </FIELD>
         <FIELD type="-98" name="SHAPEFIELD">
           <FIELDVALUE>
                <POINT x="-70.147315455731" y="44.102606073962" />
           </FIELDVALUE>
        </FIELD>
          <FIELD type="12" name="House Range" size="3" precision="0">
            <FIELDVALUE valuestring="162" />
        </FIELD>
          <FIELD type="12" name="Name" size="9" precision="0">
            <FIELDVALUE valuestring="Golder Rd" />
        </FIELD>
          <FIELD type="12" name="Postal Code" size="5" precision="0">
            <FIELDVALUE valuestring="04240" />
        </FIELD>
          <FIELD type="12" name="City" size="8" precision="0">
             <FIELDVALUE valuestring="Lewiston" />
        </FIELD>
          <FIELD type="12" name="State or Province" size="2" precision="0">
            <FIELDVALUE valuestring="ME" />
        </FIELD>
          <FIELD type="12" name="Country" size="2" precision="0">
            <FIELDVALUE valuestring="US" />
        </FIELD>
      </FEATURE>
      <GCCOUNT count="1" />
    </GEOCODE>
</RESPONSE>
</ARCXML>
```

GET_DRIVETIME

Used in: REQUEST
Parent elements: REQUEST

```
<GET_DRIVETIME>
   <DTPROPERTIES... />
  <DTCENTER... />
   <DTZONES... />
  <FEATURECOORDSYS... />
 <FILTERCOORDSYS... />
</GET_DRIVETIME >
```

Bold: Attribute or child element is required.

Description:

Request to get accessibility zones (drive time zones)

Restrictions:

None

Notes:

See DRIVETIME for response.

Drive time requests must be routed to the Route 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=9.1

&ServiceName=myservice

&CustomService=Route

&Form=True&Encode=True

Attribute descriptions for ENVELOPE:

Attribute	Usage
maxx	Right top x-coordinate in map units.

Example for GET_DRIVETIME:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
<REQUEST>
<GET_DRIVETIME>
<DTPROPERTIES>
<DTTYPE>quickest</DTTYPE>
<SPEED_FACTOR>0.8607871720116618</SPEED_FACTOR>
</DTPROPERTIES>
<DTCENTER>
<POINT x="-80.21351318359315" y="25.7802246093748"/>
</DTCENTER>
<DTZONES>
<DTZONEINPUT limit="15" />
<DTZONEINPUT limit="30" />
<DTZONEINPUT limit="60" />
</DTZONES>
<FEATURECOORDSYS id="54030" />
</GET_DRIVETIME>
</REQUEST>
</ARCXML>
```

GET_GEOCODE

Used in: REQUEST
Parent element: REQUEST

```
<GET_GEOCODE
   maxcandidates ="integer"  [20]
   minscore ="0 – 100"  [60]
  spellingsensitivity ="0 – 100" [80]
 >
    <ADDRESS... />
    <LAYER... />
</GET_GEOCODE >
```

Bold: Attribute or child element is required.

Description:

Sends a request containing address information or a point.

Restrictions:

None

Notes:

- See GEOCODE for response.

- Geocode requests must be routed to the SDCGeocode 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=9.1
&ServiceName=myservice
&CustomService=SDCGeocode
&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.

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.
spellingsensitivity	A degree of tolerance to the spelling errors. The more sensitivity value the more exact search for the matching candidates will be performed. If not specified the value is to be equal to 80.

Example for GET_GEOCODE:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_GEOCODE maxcandidates="10" minscore="60" spellingsensitivity=”90">
      <LAYER id="18" />
      <ADDRESS>
        <GCTAG id="Street Address" value="380 new york St" />
        <GCTAG id="5-digit ZIP code” value="92373" />
        <GCTAG id="City" value="redlands" />
        <GCTAG id="State" value="ca" />
      </ADDRESS>
      </GET_GEOCODE>
  </REQUEST>
</ARCXML>
```

GET_ROUTE

Used in: REQUEST
Parent element: REQUEST

```
<GET_ROUTE
        name ="string" >
        <ROUTE_PROPERTIES... />
        <STOPSLIST... />
        <ROUTEFILTER... />
        <BARRIERSLIST... />
        <FEATURECOORDSYS... />
        <FILTERCOORDSYS... />
    </GET_ROUTE >
```

Bold: Attribute or child element is required

Description:

Request for driving directions and a route shape.

Restrictions:

None

Notes:

See ROUTE for response.

Attribute descriptions for GET_ROUTE:

Attribute	Usage
name	Name of route.

Example for GET_ROUTE:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_ROUTE name="Barriers">
     <STOPSLIST>
       <ROUTE_STOP id="Stop0" desc="Start2">
         <POINT x="-97.574200" y="35.460509"/>
       </ROUTE_STOP>
       <ROUTE_STOP id="Stop1" desc="Finish2">
         <POINT x="-75.324655" y="40.063945"/>
       </ROUTE_STOP>
     </STOPSLIST>
   <BARRIERSLIST>
     <BARRIER_POINT>
       <POINT x="-90.212855" y="38.611982"/>
     </BARRIER_POINT>
     <BARRIER_POINT>
       <POINT x="-89.908691" y="38.744964"/>
     </BARRIER_POINT>
   </BARRIERSLIST>
   <ROUTEFILTER>
    <ENVELOPE minx="-180.0063613231552" miny="11.358778625954183" maxx="-64.99363867684477"
maxy="78.64122137404581"/>
   </ROUTEFILTER>
   <ROUTE_PROPERTIES>
        <ROUTETYPE>shortest</ROUTETYPE>
        <HWYPREF>50</HWYPREF>
       <PRECISION>0.1463266191428886</PRECISION>
       <LOCALE language="en" country="US" variant="" />
       <LENGTH_UNITS name="kilometer"/>
    </ROUTE_PROPERTIES>
    <FEATURECOORDSYS id="21893"/>
    <FILTERCOORDSYS id="21893"/>
   </GET_ROUTE>
 </REQUEST>
</ARCXML>
```

GET_SERVICE_INFO

Used in: REQUEST
Parent element: REQUEST

<GET_SERVICE_INFO>
No attributes
No child elements
</GET_SERVICE_INFO >

Description:

Requests information about a Route Service.

Restrictions:

None

Notes:

See SERVICEINFO for response.

Example for GET_SERVICE_INFO:

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

HWYPREF

Used in: REQUEST
Parent element: ROUTE_PROPERTIES

<HWYPREF >
No attributes
No child elements
</HWYPREF >

Description:

Defines the preference for using highways.

Restrictions:

None

Notes:

Highway preference is defined by a value ranging from 0 to 100. The value of 0 means to avoid highways, and the value of 100 means to use highways whenever possible. The default value is 50.

<HWYPREF>50**</HWYPREF>**

Example for HWYPREF:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_ROUTE name="myroute”>
      <STOPSLIST>
        <ROUTE_STOP id="0" desc="75080">
```

```
          <POINT x="-96.743072" y="32.971216"/>
        </ROUTE_STOP>
        <ROUTE_STOP id="1" desc="75081">
          <POINT x="-96.710713" y="32.948611"/>
        </ROUTE_STOP>
      </STOPSLIST>
      <ROUTEFILTER>
        <ENVELOPE minx="-180.0063613231552" miny="11.358778625954183" maxx="-64.99363867684477"
maxy="78.64122137404581"/>
      </ROUTEFILTER>
      <ROUTE_PROPERTIES>
        <ROUTETYPE>quickest</ROUTETYPE>
         <HWYPREF>50</HWYPREF>
        <PRECISION>0.1463266191428886</PRECISION>
       <LOCALE language="en" country="US" variant="" />
      <LENGTH_UNITS name="kilometer"/>
    </ROUTE_PROPERTIES>
    </GET_ROUTE>
  </REQUEST>
</ARCXML>
```

LAYER

Used in: REQUEST
Parent element: GET_GEOCODE

<LAYER
When parent element is ***GET_GEOCODE:***
id =“*string*”
When parent element is ***MAP:***
id =“*string*”
type =“*featureclass* | *acetate* | *image*”
name =“*string*”
visible =“*true* | *false*” **[true]**
maxscale =“*string*”
minscale =“*string*”
>
When parent element is ***GET_GEOCODE:****No child elements*
When parent element is ***MAP:***
<DATASET... />
<STREETIDFIELD…/>
(m) <EXTENSION... />
</LAYER >

Bold: Attribute or child element is required.

Description:

- Used in a GET_GEOCODE request to identify which layer is to be geocoded.
- Used in MAP to define SDC layers.

Restrictions:

ID for each layer must be unique within the service and within any layers added dynamically in a request.

Notes:

None

Attribute descriptions for LAYER:

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.

Attribute descriptions for LAYER when the parent element is MAP:

Attribute	Usage
id	Unique ID for a layer. The ID can be any combination of alpha and numeric characters.
name	Layer name. Can be an alias.
type	Specifies layer type. Use “featureclass” for shapefiles, SDC layers, 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.
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.

Examples for LAYER:

Example1 : Layer used in request

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_GEOCODE maxcandidates="20" minscore="60">
      <LAYER id="18" />
        <ADDRESS>
          <GCTAG id="Street Address" value="380 new york St" />
          <GCTAG id="5-digit ZIP code" value="92373" />
          <GCTAG id="City" value="redlands" />
          <GCTAG id="State" value="ca" />
        </ADDRESS>
     </GET_GEOCODE>
  </REQUEST>
</ARCXML>
```

Example 2: Layer used in **CONFIG**

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
     <UIFONT color="0,0,0" name="Arial” size="12" style="regular"/>
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
       <MAPUNITS units="decimal_degrees"/>
      </PROPERTIES>
      <WORKSPACES>
       <SDCWORKSPACE name="sws" directory="C:\data\GDT"/>
       </WORKSPACES>
       <LAYER type="featureclass" name="World Grid" visible="true" id="0">
       <DATASET name="Grid" type="polygon" workspace="sws"/>
      </LAYER>
       <LAYER type="featureclass" name="Reverse Address” visible="false" id="ReverseAddress"
```

```
maxscale="1:1">
        <DATASET name="streets" type="line" workspace="sws"/>
        <EXTENSION type="SDCGeocode">
             <GCSTYLE name="SDC.DISP_LINE_STREET_REVERSE.NA">
        <GCFIELD id="SHAPE" name="BUS_FID" />
        <GCFIELD id="PREFIX" name="PREFIX" />
        <GCFIELD id="PRETYPE" name="PRETYPE" />
        <GCFIELD id="NAME" name="NAME" />
        <GCFIELD id="TYPE" name="TYPE" />
        <GCFIELD id="SUFFIX" name="SUFFIX" />
        <GCFIELD id="L_F_ADD" name="L_F_ADD_INT" />
        <GCFIELD id="L_T_ADD" name="L_T_ADD_INT" />
        <GCFIELD id="R_F_ADD" name="R_F_ADD_INT" />
        <GCFIELD id="R_T_ADD" name="R_T_ADD_INT" />
        <GCFIELD id="ZIPL" name="POSTAL_L" />
               <GCFIELD id="ZIPR" name="POSTAL_R" />
        <GCFIELD id="CITYL" name="PLACENAME_L" />
        <GCFIELD id="CITYR" name="PLACENAME_R" />
        <GCFIELD id="STATEL" name="STATE_L" />
        <GCFIELD id="STATER" name="STATE_R" />
            </GCSTYLE>
        </EXTENSION>
      </LAYER>
      ...
  </CONFIG>
</ARCXML>
```

LAYERINFO

Used in: RESPONSE
Parent element: SERVICEINFO

```
<LAYERINFO
    id =“string”
    name =“string”
>
    <EXTENSION... />
</LAYERINFO >
```

Bold: Attribute or child element is required.

Description:

Provides the framework for information about a layer in a SERVICEINFO response.

Restrictions:

None

Notes:

None

Attribute descriptions for LAYERINFO:

Attribute	Usage
id	Reference to unique layer ID as defined in map configuration file.
name	Layer name as defined in map configuration file.

Example for LAYERINFO:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <SERVICEINFO>
      <LAYERINFO type="featureclass" name="Street Address" id="StreetAddress" >
        <FCLASS type="line"/>
        <EXTENSION type="SDCGeocode">
          <GCSTYLE name="SDC.STREET_ADDRESS.USA" >
           <GCINPUT id="Address_or_Intersection" label="Address or Intersection" width="255"
type="text" description="Address or Intersection" />
           <GCINPUT id="ZIP" label="5-digit ZIP Code" width="5" type="text" description="5-digit ZIP
Code" />
           <GCINPUT id="City" label="City" width="255" type="text" description="City" />
           <GCINPUT id="State" label="State" width="2" type="text" description="State" />
          </GCSTYLE>
        </EXTENSION>
       </LAYERINFO>
    </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

LENGTH_UNITS

Used in: REQUEST
Parent element: ROUTE_PROPERTIES

<LENGTH_UNITS
name =“*mile* | *kilometer*” [mile]
>
No child elements
</LENGTH_UNITS>

[ ]: Default value.

Description:

Units for calculating distance for driving directions.

Restrictions:

None

Notes:

None

Attribute descriptions for LENGTH_UNITS:

Attribute	Usage
name	Name of driving directions length unit.

Example for LENGTH_UNITS:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
<REQUEST>
<GET_ROUTE name="My_route">
  <STOPSLIST>
    <ROUTE_STOP id="Stop0" desc="Stop0">
      <POINT x="-117.1836" y="34.0438"/>
    </ROUTE_STOP>
    <ROUTE_STOP id="Stop1" desc="Stop1">
      <POINT x="-117.1673" y="34.0702"/>
    </ROUTE_STOP>
  </STOPSLIST>
  <ROUTE_PROPERTIES>
    <ROUTETYPE>quickest</ROUTETYPE>
    <HWYPREF>50</HWYPREF>
    <PRECISION>0.1463266191428886</PRECISION>
    <LOCALE language="en" country="US" variant="" />
     <LENGTH_UNITS name="kilometer"/>
  </ROUTE_PROPERTIES>
</GET_ROUTE>
</REQUEST>
</ARCXML>
```

LOCALE

Used in: CONFIG, REQUEST
Parent element: ENVIRONMENT, ROUTE_PROPERTIES

```
<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. In case LOCALE was not specified in CONFIG or in the request, the system locale of the machine where Route Server is installed is used.

Restrictions:

None

Notes:

- Language and country specified in the LOCALE can differ from the locale set in the system. For example, in case English locale is set in the system, and requested locale is German, then driving instructions will come back in German.

- Response will also respect decimal separator associated with requested locale, for example, if French locale is requested, coordinates in response will contain comma as decimal separator.

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”.

Example for LOCALE:

Example1 : When LOCALE is set in **REQUEST**

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_ROUTE name="myroute”>
      <STOPSLIST>
        <ROUTE_STOP id="0" desc="75080">
          <POINT x="-96.743072" y="32.971216"/>
        </ROUTE_STOP>
        <ROUTE_STOP id="1" desc="75081">
          <POINT x="-96.710713" y="32.948611"/>
        </ROUTE_STOP>
      </STOPSLIST>
      <ROUTEFILTER>
        <ENVELOPE minx="-180.0063613231552" miny="11.358778625954183" maxx="-64.99363867684477"
maxy="78.64122137404581"/>
      </ROUTEFILTER>
      <ROUTE_PROPERTIES>
        <ROUTETYPE>quickest</ROUTETYPE>
        <HWYPREF>50</HWYPREF>
```

```
        <PRECISION>0.1463266191428886</PRECISION>
          <LOCALE language="en" country="US" variant="" />
       <LENGTH_UNITS name="kilometer"/>
    </ROUTE_PROPERTIES>
    </GET_ROUTE>
  </REQUEST>
</ARCXML>
```

Example 2: When LOCALE is set 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"/>
      </ENVIRONMENT>
      ...
   </CONFIG>
</ARCXML>
```

OPTIMIZE_STOPS

Used in: REQUEST
Parent element: ROUTE_PROPERTIES

```
<OPTIMIZE_STOPS
    enable =“true | false” [false]
>
</OPTIMIZE_STOPS >
```

Bold: Attribute or child element is required.

Description:

Optimizes the order of intermediate routing stops between the Start and Finish stops.

Restrictions:

None

Notes:

None

Attribute descriptions for LOCALE:

Attribute	Usage
enable	Enables stops order optimization

Example for LOCALE:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
<REQUEST>
<GET_ROUTE name="Router”>
  <STOPSLIST>
    <ROUTE_STOP id="Stop0" desc="Stop0">
      <POINT x="-81.7302" y="41.4624"/>
    </ROUTE_STOP>
    <ROUTE_STOP id="Stop1" desc="Stop1">
      <POINT x="-81.7183" y="41.4587"/>
    </ROUTE_STOP>
    <ROUTE_STOP id="Stop2" desc="Stop2">
      <POINT x="-81.7302" y="41.4670"/>
    </ROUTE_STOP>
    <ROUTE_STOP id="Stop3" desc="Stop3">
      <POINT x="-81.7149" y="41.4599"/>
    </ROUTE_STOP>
    <ROUTE_STOP id="Stop4" desc="Stop4">
      <POINT x="-81.7235" y="41.4682"/>
    </ROUTE_STOP>
    <ROUTE_STOP id="Stop5" desc="Stop5">
      <POINT x="-81.7165" y="41.4697"/>
    </ROUTE_STOP>
  </STOPSLIST>
  <ROUTE_PROPERTIES>
    <ROUTETYPE>quickest</ROUTETYPE>
    <HWYPREF>50</HWYPREF>
    <PRECISION>0.1463266191428886</PRECISION>
    <LOCALE language="en" country="US" variant="" />
      <OPTIMIZE_STOPS enable="true" />
    <LENGTH_UNITS name="kilometer"/>
  </ROUTE_PROPERTIES>
</GET_ROUTE>
</REQUEST>
</ARCXML>
```

PART

Used in: RESPONSE
Parent element: PARTS

```
<PART
   fromstop =“string”
    tostop =“string”
 >
   (m) <SEGMENT... />
</PART >
```

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Contains the IDs for the beginning and ending points between two stops. A PART is made up of one or more segments.

Restrictions:

None

Notes:

None

Attribute descriptions for PART:

Attribute	Usage
fromstop	Stop ID of the part origin.
tostop	Stop ID of the part destination.

Example for PART:

```
<ARCXML version="1.1">
<RESPONSE>
<ROUTE name="">
  <ENVELOPE minx="-118.366388" miny="34.164836" maxx="-118.332836" maxy="34.175727"/>
  <PARTS>
   <PART fromstop="0" tostop="1" >
    <SEGMENT distance="0.000000" time="0.000000">
     <ENVELOPE minx="-118.366388" miny="34.175727" maxx="-118.366388" maxy="34.175727"/>
     <DESCRIPTION type="depart">Depart Lat: 34.17556482872054
Long:-118.36638757243382</DESCRIPTION>
      <POLYLINE>
       <PATH>
         <COORDS>-118.366388 34.175727;-118.366388 34.175727</COORDS>
        </PATH>
      </POLYLINE>
    </SEGMENT>
    </PART>
 </PARTS>
 <TOTALS distance="4.000668" time="8.433333">
    <DESCRIPTION type="totals">Driving distance: 4 mile(s)</DESCRIPTION>
    <DESCRIPTION type="totals">Driving time: 8 minute(s)</DESCRIPTION>
 </TOTALS>
</ROUTE>
</RESPONSE>
</ARCXML>
```

PARTS

Used in: RESPONSE
Parent element: ROUTE

<PARTS >

No attributes

(m) <PART... />

</PARTS >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Main element for describing parts of a route.

Restrictions:

None

Notes:

None

Example for PARTS:

```
<ARCXML version="1.1">
<RESPONSE>
<ROUTE name="">
  <ENVELOPE minx="-118.366388" miny="34.164836" maxx="-118.332836"
maxy="34.175727"/>
   <PARTS>
```

```
   <PART fromstop="0" tostop="1" >
    <SEGMENT distance="0.000000" time="0.000000">
     <ENVELOPE minx="-118.366388" miny="34.175727" maxx="-118.366388" maxy="34.175727"/>
     <DESCRIPTION type=”depart”>Depart Lat: 34.17556482872054
Long:-118.36638757243382</DESCRIPTION>
      <POLYLINE>
       <PATH>
         <COORDS>-118.366388 34.175727;-118.366388 34.175727</COORDS>
        </PATH>
      </POLYLINE>
     </SEGMENT>
    </PART>
  </PARTS>
 <TOTALS distance="4.000668" time="8.433333">
    <DESCRIPTION type="totals">Driving distance: 4 mile(s)</DESCRIPTION>
    <DESCRIPTION type="totals">Driving time: 8 minute(s)</DESCRIPTION>
 </TOTALS>
</ROUTE>
</RESPONSE>
</ARCXML>
```

PATH

Used in: RESPONSE
Parent element: POLYLINE

<PATH >
No attributes
<COORDS... />
</PATH >

Bold: Attribute or child element is required.

Description:

Represents data for a polyline feature.

Restrictions:

None

Notes:

None

Example for PATH:

```
<ARCXML version=”1.1">
<RESPONSE>
<ROUTE name="">
  <ENVELOPE minx="-118.366388" miny="34.164836" maxx=”-118.332836"
maxy="34.175727"/>
  <PARTS>
```

```
  <PART fromstop="0" tostop="1" >
   <SEGMENT distance="0.000000" time="0.000000">
    <ENVELOPE minx="-118.366388" miny="34.175727" maxx="-118.366388" maxy="34.175727"/>
    <DESCRIPTION type="depart">Depart Lat: 34.17556482872054
Long:-118.36638757243382</DESCRIPTION>
     <POLYLINE>
       <PATH>
         <COORDS>-118.366388 34.175727;-118.366388 34.175727</COORDS>
       </PATH>
     </POLYLINE>
    </SEGMENT>
   </PART>
 </PARTS>
 <TOTALS distance="4.000668" time="8.433333">
   <DESCRIPTION type="totals”>Driving distance: 4 mile(s)</DESCRIPTION>
   <DESCRIPTION type="totals”>Driving time: 8 minute(s)</DESCRIPTION>
 </TOTALS>
</ROUTE>
</RESPONSE>
</ARCXML>
```

POINT

Used in: REQUEST, RESPONSE
Parent elements: ROUTE_STOP, BARRIER_POINT, FIELDVALUE

```
<POINT
    x =“double”
     y =“double”
>
    No child elements
</POINT >
```

Bold: Attribute or child element is required.

Description:

Defines the x, y coordinate location.

Restrictions:

None

Notes:

None

Attribute descriptions for POINT:

Attribute	Usage
x	x-coordinate for point.
y	y-coordinate for point.

Examples for POINT:

Example 1: When identifying the location of a route stop in a **GET_ROUTE** request.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_ROUTE name="myroute”>
      <STOPSLIST>
        <ROUTE_STOP id="0" desc="75080">
          <POINT x="-96.743072" y="32.971216"/>
        </ROUTE_STOP>
        <ROUTE_STOP id="1" desc="75081">
          <POINT x="-96.710713" y="32.948611"/>
        </ROUTE_STOP>
      </STOPSLIST>
      <BARRIERSLIST>
        <BARRIER_POINT>
             <POINT x="-90.212855" y="38.611982"/>
        </BARRIER_POINT>
        <BARRIER_POINT>
             <POINT x="-89.908691" y="38.744964"/>
        </BARRIER_POINT>
      </BARRIERSLIST>
      <ROUTEFILTER>
        <ENVELOPE minx="-180.0063613231552" miny=”11.358778625954183" maxx="-64.99363867684477"
maxy=”78.64122137404581"/>
      </ROUTEFILTER>
      <ROUTE_PROPERTIES>
        <ROUTETYPE>quickest</ROUTETYPE>
        <HWYPREF>50</HWYPREF>
        <PRECISION>0.1463266191428886</PRECISION>
       <LOCALE language="en" country="US" variant="" />
       <LENGTH_UNITS name="kilometer"/>
      </ROUTE_PROPERTIES>
    </GET_ROUTE>
  </REQUEST>
</ARCXML>
```

Example 2: When identifying the location of an address 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=”34" precision=”0">
         <FIELDVALUE valuestring=”162 Golder Rd, 04240, ME, Lewiston” />
      </FIELD>
      <FIELD type=”-98" name=”SHAPEFIELD”>
         <FIELDVALUE>
            <POINT x=”-70.147315455731" y=”44.102606073962" />
         </FIELDVALUE>
      </FIELD>
      <FIELD type=”12" name=”House Range” size=”3" precision=”0">
         <FIELDVALUE valuestring=”162" />
      </FIELD>
      <FIELD type=”12" name=”Name” size=”9" precision=”0">
         <FIELDVALUE valuestring=”Golder Rd” />
      </FIELD>
      <FIELD type=”12" name=”Postal Code” size=”5" precision=”0">
         <FIELDVALUE valuestring=”04240" />
      </FIELD>
      <FIELD type=”12" name=”City” size=”8" precision=”0">
         <FIELDVALUE valuestring=”Lewiston” />
      </FIELD>
      <FIELD type=”12" name=”State or Province” size=”2" precision=”0">
         <FIELDVALUE valuestring=”ME” />
      </FIELD>
      <FIELD type=”12" name=”Country” size=”2" precision=”0">
         <FIELDVALUE valuestring=”US” />
      </FIELD>
    </FEATURE>
    <GCCOUNT count=”1" />
  </GEOCODE>
</RESPONSE>
</ARCXML>
```

POLYGON

Used in: RESPONSE
Parent elements: DTZONERESULT

```
<POLYGON
    coords =“x1 y1;…xn yn”
>
</POLYGON >
```

Bold: Attribute or child element is required.

Description:

Defines a polygon for the found drive time zone

Restrictions:

The polygon must be closed - the beginning and ending coordinate pairs must be identical.

Notes:

None

Example for POLYGON:

```
<?xml version=”1.0" encoding=”UTF-8"?>
<ARCXML version=”1.1">
<RESPONSE>
  <DRIVETIME>
    <DTZONERESULT id=”1" limit=”0.100000">
     <POLYGON coords=”-82.940256 33.113230;-82.939382 33.113473;-82.940256 33.113230;-82.940741
33.113080;-82.941047 33.112975;-82.941269 33.112887;-82.941097 33.112960;-82.940546 33.113149;-
82.940256 33.113230" />
    </DTZONERESULT>
   <ENVELOPE minx=”-82.941269" miny=”33.112887" maxx=”-82.939382" maxy=”33.113473"/>
  </DRIVETIME>
</RESPONSE>
</ARCXML>
```

POLYLINE

Used in: RESPONSE
Parent element: SEGMENT

<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:

None

Example for POLYLINE:

```
<ARCXML version="1.1">
<RESPONSE>
<ROUTE name="">
  <ENVELOPE minx="-118.366388" miny="34.164836" maxx="-118.332836" maxy="34.175727"/>
   <PARTS>
```

```
   <PART fromstop="0" tostop="1" >
    <SEGMENT distance="0.000000" time="0.000000">
     <ENVELOPE minx="-118.366388" miny="34.175727" maxx="-118.366388" maxy="34.175727"/>
     <DESCRIPTION type="depart”>Depart Lat: 34.17556482872054
Long:-118.36638757243382</DESCRIPTION>
       <POLYLINE>
       <PATH>
         <COORDS>-118.366388 34.175727;-118.366388 34.175727</COORDS>
        </PATH>
       </POLYLINE>
    </SEGMENT>
   </PART>
 </PARTS>
 <TOTALS distance="4.000668" time="8.433333">
    <DESCRIPTION type="totals">Driving distance: 4 mile(s)</DESCRIPTION>
    <DESCRIPTION type="totals">Driving time: 8 minute(s)</DESCRIPTION>
 </TOTALS>
</ROUTE>
</RESPONSE>
</ARCXML>
```

PRECISION

Used in: REQUEST
Parent element: ROUTE_PROPERTIES

<PRECISION >

No attributes

No child elements

</PRECISION >

Description:

Provides a way to set precision for shape generalization to minimize the response size.

Restrictions:

None

Notes:

The precision is calculated based on the maximum allowed deviation of coordinates during shape generalization. The larger the value, the more generalized the shape is. If PRECISION is not used, the resulting route shape is not generalized.

<PRECISION>0.1463266191428886**</PRECISION>**

Example for PRECISION:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_ROUTE name="myroute”>
      <STOPSLIST>
```

```
        <ROUTE_STOP id="0" desc="75080">
          <POINT x="-96.743072" y="32.971216"/>
        </ROUTE_STOP>
        <ROUTE_STOP id="1" desc="75081">
          <POINT x=”-96.710713" y=”32.948611"/>
        </ROUTE_STOP>
      </STOPSLIST>
      <ROUTEFILTER>
        <ENVELOPE minx="-180.0063613231552" miny="11.358778625954183" maxx="-64.99363867684477"
maxy="78.64122137404581"/>
      </ROUTEFILTER>
      <ROUTE_PROPERTIES>
        <ROUTETYPE>quickest</ROUTETYPE>
        <HWYPREF>50</HWYPREF>
         <PRECISION>0.1463266191428886</PRECISION>
       <LOCALE language="en" country="US" variant=""/>
       <LENGTH_UNITS name=”kilometer”/>
      </ROUTE_PROPERTIES>
    </GET_ROUTE>
  </REQUEST>
</ARCXML>
```

REQUEST

Used in: REQUEST
Parent element: ARCXML

<REQUEST >
No attributes
<GET_DRIVETIME... /> *[Or]*
<GET_GEOCODE... /> *[Or]*
<GET_ROUTE... /> *[Or]*
<GET_SERVICE_INFO... />
</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_ROUTE response is ROUTE.
- GET_GEOCODE response is GEOCODE.
- GET_DRIVETIME response is DRIVETIME.

Example for REQUEST:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_ROUTE name="myroute”>
      <STOPSLIST>
        <ROUTE_STOP id="0" desc="75080">
          <POINT x="-96.743072" y="32.971216"/>
        </ROUTE_STOP>
        <ROUTE_STOP id="1" desc="75081">
          <POINT x="-96.710713" y="32.948611"/>
        </ROUTE_STOP>
      </STOPSLIST>
      <ROUTEFILTER>
        <ENVELOPE minx="-180.0063613231552" miny="11.358778625954183" maxx="-64.99363867684477"
maxy="78.64122137404581"/>
      </ROUTEFILTER>
      <ROUTE_PROPERTIES>
        <ROUTETYPE>quickest</ROUTETYPE>
        <HWYPREF>50</HWYPREF>
        <PRECISION>0.1463266191428886</PRECISION>
       <LOCALE language="en" country="US" variant="" />
       <LENGTH_UNITS name="kilometer"/>
      </ROUTE_PROPERTIES>
    </GET_ROUTE>
  </REQUEST>
</ARCXML>
```

RESPONSE

Used in: RESPONSE
Parent element: ARCXML

<RESPONSE >
No attributes
<DRIVETIME... /> *[Or]*
<GEOCODE... /> *[Or]*
<ROUTE... /> *[Or]*
<SERVICEINFO... />
</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.
- ROUTE is the response to GET_ROUTE.
- GEOCODE is the response to GET_GEOCODE.
- DRIVETIME is the response to GET_ DRIVETIME.

Example for RESPONSE:

```
<ARCXML version="1.1">
<RESPONSE>
<ROUTE name="">
  <ENVELOPE minx="-118.366388" miny="34.164836" maxx="-118.332836"maxy="34.175727"/>
  <PARTS>
   <PART fromstop="0" tostop="1" >
    <SEGMENT distance="0.000000" time="0.000000">
     <ENVELOPE minx="-118.366388" miny="34.175727" maxx="-118.366388" maxy="34.175727"/>
     <DESCRIPTION type="depart">Depart Lat: 34.17556482872054 Long:-118.36638757243382</DESCRIPTION>
      <POLYLINE>
        <PATH>
          <COORDS>-118.366388 34.175727;-118.366388 34.175727</COORDS>
         </PATH>
      </POLYLINE>
     </SEGMENT>
    </PART>
 </PARTS>
 <TOTALS distance="4.000668" time="8.433333">
    <DESCRIPTION type="totals”>Driving distance: 4 mile(s)</DESCRIPTION>
    <DESCRIPTION type="totals”>Driving time: 8 minute(s)</DESCRIPTION>
 </TOTALS>
</ROUTE>
</RESPONSE>
</ARCXML>
```

RESTRICTION

Used in: RESPONSE
Parent elements: RESTRICTIONSLIST

```
<RESTRICTION
    name =“string”
    type =“boolean|minallowed|maxallowed”
    setbydefault=“true | false”
 >
</RESTRICTION >
```

Bold: Attribute or child element is required.

Description:

Describes a route restriction: its name and type

Restrictions:

None

Notes:

If the restriction name contains special XML characters then the ArcIMS server will return this name being converted into valid XML string. E.g. the restriction name “Don’t Use Bridges” wil be returned as “Don't Use Bridges”.

Attribute descriptions for RESTRICTION:

Attribute	Usage
name	Name of the restriction in the dataset
type	Type of the restriction; it can be boolean (yes/no type), minallowed, or maxallowed.
setbydefault	Shows whether a restriction is set or not if neither SETRESTRICTION nor CLEARRESTRICTION are specified in a request

Example for RESTRICTION:

```
<?xml version=”1.0" encoding=”UTF-8"?>
<ARCXML version=”1.1">
<RESPONSE>
 <SERVICEINFO>
  <LAYERINFO type=”featureclass” id=”Layer6" name=”Streets” >
   <FCLASS type=”line” />
    <EXTENSION type=”Route”>
     <RESTRICTIONSLIST>
       <RESTRICTION name=”Carpool” type=”boolean” setbydefault=”false” />
       <RESTRICTION name=”Pedestrian” type=”boolean” setbydefault=”false” />
       <RESTRICTION name=”Height” type=”maxallowed” setbydefault=”false” />
       <RESTRICTION name=”Weight” type=”maxallowed” setbydefault=”false” />
       <RESTRICTION name=”TurnRestriction” type=”boolean” setbydefault=”true”/>
       <RESTRICTION name=”OneWay” type=”boolean” setbydefault=”true”/>
     </RESTRICTIONSLIST>
    </EXTENSION>
  </LAYERINFO>
 </SERVICEINFO>
</RESPONSE>
</ARCXML>
```

RESTRICTIONS

Used in: REQUEST
Parent elements: GET_ROUTE

<RESTRICTIONS>
(m)<SETRESTRICTION.../>
(m) <CLEARRESTRICTION... />

</RESTRICTIONS >

(m): Child element can be used multiple times.

Description:

It is a parent element for route restrictions.

Restrictions:

None

Notes:

None

Example for RESTRICTIONS:

```
<?xml version=”1.0" encoding=”UTF-8" ?>
<ARCXML version=”1.1">
<REQUEST>
  <GET_ROUTE name=”RT_002">
    <STOPSLIST>
```

```
    <ROUTE_STOP id=”0" desc=”Lat: 38.47  Long:-121.51">
      <POINT x=”-121.51" y=”38.47"/>
    </ROUTE_STOP>
    <ROUTE_STOP id=”1" desc=”Lat: 38.49011585133329 Long: -121.45621949173282">
      <POINT x=”-121.45621949173282" y=”38.49011585133329"/>
      </ROUTE_STOP>
    <ROUTE_STOP id=”2" desc=”Lat: 38.48740970639922 Long: -121.45949914879973">
      <POINT x=”-121.45949914879973" y=”38.48740970639922"/>
      </ROUTE_STOP>
   </STOPSLIST>
  <ROUTE_PROPERTIES>
      <RESTRICTIONS>
     <SETRESTRICTION name=”Height” parameter = “5” type=”strict” />
     <SETRESTRICTION name=”CarPool” type=”strict” />
     <CLEARRESTRICTION name=”OneWay” />
      </RESTRICTIONS>
   <ROUTETYPE>quickest</ROUTETYPE>
    <PRECISION>0.0005</PRECISION>
  </ROUTE_PROPERTIES>
 </GET_ROUTE>
</REQUEST>
</ARCXML>
```

RESTRICTIONSLIST

Used in: RESPONSE
Parent elements: EXTENSION

```
<RESTRICTIONSLIST>
   <m><RESTRICTION.../>
</RESTRICTIONSLIST >
```

Description:

It is an element containing all restrictions, which are configured for the layer.

Restrictions:

None

Notes:

None

Example for RESTRICTIONS:

```
<?xml version=”1.0" encoding=”UTF-8"?>
<ARCXML version=”1.1">
<RESPONSE>
  <SERVICEINFO>
    <LAYERINFO type=”featureclass” id=”Layer6" name=”Streets” >
      <FCLASS type=”line” />
      <EXTENSION type=”Route”>
```

```
        <RESTRICTIONSLIST>
       <RESTRICTION name="Carpool" type="boolean" />
       <RESTRICTION name="Pedestrian" type="boolean" />
       <RESTRICTION name="Height" type="maxallowed" />
       <RESTRICTION name="Weight" type="maxallowed" />
        </RESTRICTIONSLIST>
    </EXTENSION>
   </LAYERINFO>
  </SERVICEINFO>
</RESPONSE>
</ARCXML>
```

ROUTE

Used in: RESPONSE
Parent element: RESPONSE

```
<ROUTE
   name =“string”
 >
   <ENVELOPE... />
    <PARTS... />
    <TOTALS... />
   <ERROR... />
</ROUTE >
```

Bold: Attribute or child element is required.

Description:

Contains information on the resulting route that was requested.

Restrictions:

The child elements returned with ROUTE are ENVELOPE, PARTS, and TOTALS. If an error is found in the route, an ERROR is returned.

Notes:

- See GET_ROUTE for request.
- The resulting route is split into segments between each stop. Each segment includes driving directions. If a route cannot be found • or an error occurs, an ERROR is returned.

Attribute descriptions for ROUTE:

Attribute	Usage
name	Name returned to identify a route. The name is the same one used in the GET_ROUTE request. If no *name* was included in the request, no *name* is included in the response.

Examples for ROUTE:

Example 1: When a route is found.

```
<ARCXML version="1.1">
<RESPONSE>
<ROUTE name="myroute”>
  <ENVELOPE minx="-118.366388" miny="34.164836" maxx="-118.332836"
maxy=”34.175727"/>
  <PARTS>
   <PART fromstop="0" tostop="1" >
    <SEGMENT distance="0.000000" time="0.000000">
     <ENVELOPE minx=”-118.366388" miny="34.175727" maxx=”-118.366388" maxy=”34.175727"/>
     <DESCRIPTION type=”depart”>Depart Lat: 34.17556482872054
Long:-118.36638757243382</DESCRIPTION>
      <POLYLINE>
        <PATH>
          <COORDS>-118.366388 34.175727;-118.366388 34.175727</COORDS>
         </PATH>
      </POLYLINE>
     </SEGMENT>
    </PART>
 </PARTS>
 <TOTALS distance="4.000668" time="8.433333">
    <DESCRIPTION type="totals">Driving distance: 4 mile(s)</DESCRIPTION>
    <DESCRIPTION type="totals">Driving time: 8 minute(s)</DESCRIPTION>
 </TOTALS>
</ROUTE>
</RESPONSE>
</ARCXML>
```

Example 2: When no route is found.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <ROUTE name="myroute" />
  </RESPONSE>
</ARCXML>
```

Example 3: When there is an error with the route.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
    <ROUTE  name="myroute">
      <ERROR machine="RouteServer" processid=”440" threadid=”398">
         Selected route type is not supported.
      </ERROR>
    </ROUTE>
  </RESPONSE>
</ARCXML>
```

ROUTE_PROPERTIES

Used in: REQUEST
Parent element: GET_ROUTE

<ROUTE_PROPERTIES >
No attributes
<ROUTETYPE... />
<HWYPREF... />
<PRECISION... />
<LOCALE…/>
<OPTIMIZE_STOPS…/>
<LENGTH_UNITS…/>
</ROUTE_PROPERTIES >

Bold: Attribute or child element is required.

Description:

Parent element for route properties.

Restrictions:

LOCALE is required when the requested language is different from the default language.

Notes:

- If the LENGTH_UNITS child element is omitted, “miles” are used as a default length unit.
- If the OPTIMIZE_STOPS child element is omitted, the order of stops is not optimized during route calculation.
- If the PRECISION element is not used, the resulting route shape is not generalized and the full geometry is returned in the response.

- If the HWYPREF child element is omitted, the highways preference defaults to 50.
- If the LOCALE element is omitted, the locale defined in the map service will be used.

Example for ROUTE_PROPERTIES:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_ROUTE name="myroute">
      <STOPSLIST>
        <ROUTE_STOP id="0" desc="75080">
          <POINT x="-96.743072" y="32.971216"/>
        </ROUTE_STOP>
        <ROUTE_STOP id="1" desc="75081">
          <POINT x="-96.710713" y="32.948611"/>
        </ROUTE_STOP>
      </STOPSLIST>
      <ROUTEFILTER>
        <ENVELOPE minx="-180.0063613231552" miny="11.358778625954183" maxx="-64.99363867684477"
maxy="78.64122137404581"/>
      </ROUTEFILTER>
       <ROUTE_PROPERTIES>
        <ROUTETYPE>quickest</ROUTETYPE>
        <HWYPREF>50</HWYPREF>
        <PRECISION>0.1463266191428886</PRECISION>
       <LOCALE language="en" country="US" variant="" />
       <OPTIMIZE_STOPS enable="false" />
       <LENGTH_UNITS name="kilometer"/>
       </ROUTE_PROPERTIES>
    </GET_ROUTE>
  </REQUEST>
</ARCXML>
```

ROUTE_STOP

Used in: REQUEST
Parent element: STOPSLIST

```
<ROUTE_STOP
    id =“string”
    desc =“string”
>
    <POINT... />
</ROUTE_STOP >
```

Bold: Attribute or child element is required.

Description:

Defines the ID and description of a routing stop.

Restrictions:

None

Notes:

None

Attribute descriptions for ROUTE_STOP:

Attribute	Usage
desc	Description of the stop that appears in the driving directions arrive/depart items. If no description is provided, a default stop description is generated.
id	Unique ID for a stop. The ID can be any combination of alpha and numeric characters.

Example for ROUTE_STOP:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_ROUTE name="myroute”>
      <STOPSLIST>
       <ROUTE_STOP id="0" desc="99 Broadway FL 12,New York, NY, 10012">
         <POINT x="-96.743072" y="32.971216"/>
       </ROUTE_STOP>
       <ROUTE_STOP id="1" desc=”12 Broadway FL 12,New York, NY, 10012">
         <POINT x="-96.710713" y=”32.948611"/>
       </ROUTE_STOP>
      </STOPSLIST>
      <ROUTEFILTER>
        <ENVELOPE minx="-180.0063613231552" miny="11.358778625954183" maxx="-64.99363867684477"
maxy="78.64122137404581"/>
      </ROUTEFILTER>
      <ROUTE_PROPERTIES>
        <ROUTETYPE>quickest</ROUTETYPE>
        <HWYPREF>50</HWYPREF>
        <PRECISION>0.1463266191428886</PRECISION>
       <LOCALE language="en" country="US" variant="" />
       <LENGTH_UNITS name="kilometer"/>
      </ROUTE_PROPERTIES>
    </GET_ROUTE>
  </REQUEST>
</ARCXML>
```

ROUTEFILTER

Used in: REQUEST
Parent element: GET_ROUTE

<ROUTEFILTER >
No attributes

<ENVELOPE... />

</ROUTEFILTER >

Bold: Attribute or child element is required.

Description:

Allows an envelope to be added to the request for faster processing.

Restrictions:

None

Notes:

The route filter sets a clipping rectangle for the returned route. In this case, only those points that are visible within the specified extent are returned in the response. However, all directions are returned.

Example for ROUTEFILTER:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_ROUTE name="myroute">
      <STOPSLIST>
        <ROUTE_STOP id="0" desc="75080">
          <POINT x="-96.743072" y="32.971216"/>
        </ROUTE_STOP>
        <ROUTE_STOP id="1" desc="75081">
          <POINT x="-96.710713" y="32.948611"/>
        </ROUTE_STOP>
      </STOPSLIST>
       <ROUTEFILTER>
        <ENVELOPE minx="-180.0063613231552" miny="11.358778625954183" maxx="-64.99363867684477"
maxy="78.64122137404581"/>
       </ROUTEFILTER>
      <ROUTE_PROPERTIES>
        <ROUTETYPE>quickest</ROUTETYPE>
        <HWYPREF>50</HWYPREF>
        <PRECISION>0.1463266191428886</PRECISION>
       <LOCALE language="en" country="US" variant="" />
       <LENGTH_UNITS name="kilometer"/>
      </ROUTE_PROPERTIES>
    </GET_ROUTE>
  </REQUEST>
</ARCXML>
```

ROUTETYPE

Used in: REQUEST
Parent element: ROUTE_PROPERTIES

<ROUTETYPE >

No attributes

No child elements

</ROUTETYPE >

Description:

Defines route searching criteria for using either the quickest or shortest route.

Restrictions:

None

Notes:

The valid content words are “quickest” and “shortest”.

<ROUTETYPE>quickest**</ROUTETYPE >**
<ROUTETYPE>shortest**</ROUTETYPE >**

Example for ROUTETYPE:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_ROUTE name="myroute”>
      <STOPSLIST>
        <ROUTE_STOP id="0" desc="75080">
          <POINT x="-96.743072" y="32.971216"/>
        </ROUTE_STOP>
        <ROUTE_STOP id="1" desc="75081">
          <POINT x="-96.710713" y="32.948611"/>
        </ROUTE_STOP>
      </STOPSLIST>
      <ROUTEFILTER>
        <ENVELOPE minx="-180.0063613231552" miny="11.358778625954183" maxx="-64.99363867684477"
maxy="78.64122137404581"/>
      </ROUTEFILTER>
      <ROUTE_PROPERTIES>
         <ROUTETYPE>quickest</ROUTETYPE>
        <HWYPREF>50</HWYPREF>
        <PRECISION>0.1463266191428886</PRECISION>
       <LOCALE language="en" country="US" variant="" />
       <LENGTH_UNITS name="kilometer"/>
     </ROUTE_PROPERTIES>
    </GET_ROUTE>
  </REQUEST>
</ARCXML>
```

SDCWORKSPACE

Used in: CONFIG
Parent element: WORKSPACES

<SDCWORKSPACE
name =“*string*”
directory =“*string*”
>
No child elements
</SDCWORKSPACE >

Bold: Attribute or child element is required.

Description:

Specifies a workspace for a SDC dataset used with the Route Server.

Restrictions:

None

Notes:

None

Attribute descriptions for SDCWORKSPACE:

Attribute	Usage
directory	Directory containing the Route Server data in SDC format. UNC pathnames can be used (`\\myComputer\sdcdirectory`).
name	Workspace name. Must be unique among all data sources.

Example for SDCWORKSPACE:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
     <UIFONT color="0,0,0" name="Arial" size="12" style="regular"/>
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
       <MAPUNITS units="decimal_degrees"/>
      </PROPERTIES>
      <WORKSPACES>
        <SDCWORKSPACE name="sws" directory="C:\data\GDT"/>
       </WORKSPACES>
       <LAYER type="featureclass" name="World Grid" visible="true" id="0">
       <DATASET name="Grid" type="polygon" workspace="sws"/>
      </LAYER>
       ...
  </CONFIG>
</ARCXML>
```

SEGMENT

Used in: RESPONSE
Parent element: PART

```
<SEGMENT
    distance ="double"
   time ="double"
    eta = "string"
>
   (m) <DESCRIPTION... />
   <ENVELOPE... />
   <POLYLINE... />
</SEGMENT >
```

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Contains time and distance values for a segment.

Restrictions:

Format of "eta" attribute is "hh:mm".

Notes:

- Distance attribute is always in miles, although directions in the text will converted to the units setting specified in the request.

- Eta attribute is reported only when TRIP_START element is specified.
- Eta is reported in **departure local time**.

Attribute descriptions for SEGMENT:

Attribute	Usage
distance	Length of the segment in miles.
time	Traversal time of the segment in minutes.
eta	Estimates time of arrival at the segment.

Examples for SEGMENT:

Example 1: Distance and **time** reporting.

```
<ARCXML version="1.1">
<RESPONSE>
<ROUTE name="">
  <ENVELOPE minx="-118.366388" miny="34.164836" maxx="-118.332836"
maxy="34.175727"/>
  <PARTS>
   <PART fromstop="0" tostop="1" >
     <SEGMENT distance="0.000000" time="0.000000">
        <ENVELOPE minx="-118.366388" miny="34.175727" maxx="-118.366388" maxy="34.175727"/>
         <DESCRIPTION type="depart”>Depart Lat: 34.17556482872054
Long:-118.36638757243382</DESCRIPTION>
          <POLYLINE>
            <PATH><COORDS>-118.366388 34.175727;-118.366388 34.175727</COORDS></PATH>
          </POLYLINE>
      </SEGMENT>
     <SEGMENT distance="1.624258" time="3.267459">
        <ENVELOPE minx="-118.360582" miny="34.164836" maxx="-118.332836" maxy="34.172164"/>
         <DESCRIPTION type="direction">Turn left on Magnolia Blvd </DESCRIPTION>
         <DESCRIPTION type="drivetext">Drive 1.6 mile(s) ~ 3 minute(s) </DESCRIPTION>
```

```
        <POLYLINE>
          <PATH><COORDS>-118.360582 34.164855;-118.355164 34.164836;-118.354345 34.164982;-118.332836
34.172164</COORDS></PATH>
        </POLYLINE>
      </SEGMENT>
        ...

     <SEGMENT distance="0.000000" time="0.000000">
        <ENVELOPE minx="-118.347496" miny="34.173101" maxx="-118.347496"
maxy="34.173101"/>
        <DESCRIPTION type="arrive">Arrive at Lat: 34.173099868
Long:-118.347498224</DESCRIPTION>
        <POLYLINE>
          <PATH><COORDS>-118.347496 34.173101;-118.347496 34.173101</COORDS></PATH>
        </POLYLINE>
      </SEGMENT>
   </PART>
 </PARTS>
 <TOTALS distance="4.000668" time="8.433333">
    <DESCRIPTION type="totals">Driving distance: 4 mile(s)
    </DESCRIPTION>
    <DESCRIPTION type="totals">Driving time: 8 minute(s)
    </DESCRIPTION>
 </TOTALS>
</ROUTE>
</RESPONSE>
</ARCXML>
```

Example 2: Eta reporting.

```
<?xml version=”1.0" encoding=”UTF-8"?>
<ARCXML version=”1.1">
<RESPONSE>
<ROUTE name=””>
  <ENVELOPE minx=”-122.494255" miny=”38.206298" maxx=”-73.427217" maxy=”45.474996"/>
  <PARTS>
   <PART fromstop=”0" tostop=”1" >
     <SEGMENT distance=”0.000000" time=”0.000000" eta=”12:00">
     <ENVELOPE minx=”-73.955177" miny=”44.518557" maxx=”-73.955177" maxy=”44.518557"/>
```

```
      <DESCRIPTION type=”depart”>Depart Start</DESCRIPTION>
      <POLYLINE>
     <PATH><COORDS>-73.955177 44.518557;-73.955177 44.518557</COORDS></PATH>
      </POLYLINE>
     </SEGMENT>
    <SEGMENT distance=”1.745241" time=”4.188485" eta=”12:00">
    <ENVELOPE minx=”-73.972379" miny=”44.518557" maxx=”-73.955177" maxy=”44.535043"/>
      <DESCRIPTION type=”direction”>Go North West on Alderbrook Rd toward Alder Brook Park Rd</
DESCRIPTION>
      <DESCRIPTION type=”drivetext”>Drive 1.7 mile(s) ~ 4 minute(s)</DESCRIPTION>
       <POLYLINE>
     <PATH><COORDS>-73.955177 44.518557;-73.966379 44.520652;-73.965361 44.523484;-73.966252
44.526273;-73.965783 44.530505;-73.972379 44.535043</COORDS></PATH>
       </POLYLINE>
     </SEGMENT>
    <SEGMENT distance=”0.214002" time=”0.513533" eta=”12:04">
    <ENVELOPE minx=”-73.972379" miny=”44.535043" maxx=”-73.970552" maxy=”44.537327"/>
      <DESCRIPTION type=”direction”>Turn right at Unnamed St to stay on Alderbrook Rd</DESCRIPTION>
      <DESCRIPTION type=”drivetext”>Drive 0.2 mile(s) ~ &lt; 1 minute</DESCRIPTION>
       <POLYLINE>
     <PATH><COORDS>-73.972379 44.535043;-73.970795 44.535985;-73.971073 44.537327</COORDS></PATH>
       </POLYLINE>
      </SEGMENT>
       ...
     <SEGMENT distance=”0.049106" time=”0.117876" eta=”20:25">
     <ENVELOPE minx=”-122.494255" miny=”38.521606" maxx=”-122.493699" maxy=”38.522167"/>
       <DESCRIPTION type=”direction”>Turn right on York Ln</DESCRIPTION>
       <DESCRIPTION type=”drivetext”>Drive &lt; 0.1 mile(s) ~ &lt; 1 minute</DESCRIPTION>
        <POLYLINE>
      <PATH><COORDS>-122.494255 38.521606;-122.493699 38.522167</COORDS></PATH>
        </POLYLINE>
      </SEGMENT>
     <SEGMENT distance=”0.000000" time=”0.000000" eta=”20:25">
     <ENVELOPE minx=”-122.493699" miny=”38.522167" maxx=”-122.493699" maxy=”38.522167"/>
       <DESCRIPTION type=”arrive”>Arrive at Finish, on the right</DESCRIPTION>
        <POLYLINE>
     <PATH><COORDS>-122.493699 38.522167;-122.493699 38.522167</COORDS></PATH>
        </POLYLINE>
      </SEGMENT>
```

```
    </PART>
    </PARTS>
  <TOTALS distance=”3051.718803" time=”3385.214602">
   <DESCRIPTION type=”totals”>Driving distance: 3051.7 mile(s)</DESCRIPTION>
   <DESCRIPTION type=”totals”>Driving time: 56 hour(s) 25 minute(s)</DESCRIPTION>
  </TOTALS>
</ROUTE>
</RESPONSE>
</ARCXML>
```

SERVICEINFO

Used in: RESPONSE
Parent element: RESPONSE

<SERVICEINFO >
No attributes

(m) <LAYERINFO... />

</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:

None

Notes:

See GET_SERVICE_INFO for request.

Example for SERVICEINFO:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <RESPONSE>
     <SERVICEINFO>
     <LAYERINFO type="featureclass" name="Street Address" id="StreetAddress" >
        <FCLASS type="line"/>
        <EXTENSION type="SDCGeocode">
          <GCSTYLE name="SDC.STREET_ADDRESS.USA" >
           <GCINPUT id="Address_or_Intersection" label="Address or Intersection" width="255"
type="text" description="Address or Intersection" />
           <GCINPUT id="ZIP" label="5-digit ZIP Code" width="5" type="text" description="5-digit ZIP
Code” />
           <GCINPUT id="City" label="City" width="255" type="text" description="City" />
           <GCINPUT id="State" label="State" width="2" type="text" description="State" />
          </GCSTYLE>
        </EXTENSION>
      </LAYERINFO>
     </SERVICEINFO>
  </RESPONSE>
</ARCXML>
```

SETRESTRICTION

Used in: REQUEST
Parent element: RESTRICTIONS

```
<SETRESTRICTION
  name = "string"
  type = "string"
  parameter = "double"
/>
</SETRESTRICTION >
```

Description:

This is an element specifying route restriction and its usage type.

Restrictions:

None

Notes:

- Usage type "relaxed" allows a restriction to be violated if there are no other ways to find restricted route. Type "strict" denies violating a restriction under any circumstances.
- "Relaxed" type has no effect on special restrictions "OneWay" and "TurnRestriction". These restrictions are always "strict".
- "Parameter" attribute must not be set for boolean restrictions and is used with "maxallowed" and "minallowed" restrictions only.
- Special XML characters in the restriction name must be changed to their string values so the Route Server could correctly interpret the XML string. E.g. change "Don't Use Bridges" to "Don't Use Briges".

Attribute descriptions for SETRESTRICTION:

Attribute	Usage
name	Name of the restriction as it is specified in the dataset (see SERVICEINFO/RESTRICTIONSLIST)
type	Type of the restriction (strict/relaxed)
parameter	Value of minallowed/maxallowed restriction

Example for SETRESTRICTION:

```
<?xml version=”1.0" encoding=”UTF-8" ?>
<ARCXML version=”1.1">
<REQUEST>
  <GET_ROUTE name=”RT_002">
   <STOPSLIST>
    <ROUTE_STOP id=”0" desc=”Lat: 38.47  Long:-121.51">
      <POINT x=”-121.51" y=”38.47"/>
    </ROUTE_STOP>
   <ROUTE_STOP id=”1" desc=”Lat: 38.49011585133329 Long: -121.45621949173282">
     <POINT x=”-121.45621949173282" y=”38.49011585133329"/>
   </ROUTE_STOP>
   <ROUTE_STOP id=”2" desc=”Lat: 38.48740970639922 Long: -121.45949914879973">
     <POINT x=”-121.45949914879973" y=”38.48740970639922"/>
   </ROUTE_STOP>
  </STOPSLIST>
   <ROUTE_PROPERTIES>
     <RESTRICTIONS>
       <SETRESTRICTION name=”Height” parameter = “5” type=”strict” />
       <SETRESTRICTION name=”CarPool” type=”strict” />
```

```
    </RESTRICTIONS>
   <ROUTETYPE>quickest</ROUTETYPE>
     <PRECISION>0.0005</PRECISION>
    </ROUTE_PROPERTIES>
  </GET_ROUTE>
</REQUEST>
</ARCXML>
```

SPEED_FACTOR

Used in: REQUEST
Parent element: DTPROPERTIES, ROUTE, PROPERTIES

<SPEED_FACTOR >

No attributes

No child elements

</SPEED_FACTOR >

Description:

This element permits adjusting of drive speed while searching routes or drive time zones.

Restrictions:

Speed factor value may be set from range 0.1…10. It allows varying of driving speed from ten times slower to ten times faster than the speed values stored in the dataset.

Notes:

Setting this element has sense only when “quickest” routes or drive time zones are being searched and affects on driving time values only.

Examples for SPEED_FACTOR:

Example 1: Using of speed factor in a GET_DRIVETIME request.

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

```
  <GET_DRIVETIME>
    <DTPROPERTIES>
      <DTTYPE>quickest</DTTYPE>
          <SPEED_FACTOR>0.861</SPEED_FACTOR>
      <DTDIRECTION>inbound</DTDIRECTION>
    </DTPROPERTIES>
    <DTCENTER>
      <POINT x="-80.21351318359315" y="25.7802246093748"/>
    </DTCENTER>
    <DTZONES>
      <DTZONEINPUT limit="15" />
      <DTZONEINPUT limit="30" />
      <DTZONEINPUT limit="60" />
    </DTZONES>
    <FEATURECOORDSYS id="54030" />
  </GET_DRIVETIME>
</REQUEST>
</ARCXML>
```

Example 2: Using of speed factor in a GET_ROUTE request.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
<REQUEST>
  <GET_ROUTE name="RT_003">
    <STOPSLIST>
      <ROUTE_STOP id="1" desc="Graphical Pick: -3743,17937898357, 412809,470786583">
        <POINT x="-96.0449754633439" y="41.1840207589678"/>
       </ROUTE_STOP>
```

```
    <ROUTE_STOP id=”2" desc=”Graphical Pick: 121872,235421235, 193658,55515471">
      <POINT x=”-94.575253499639" y=”39.2196744329457"/>
     </ROUTE_STOP>
    <ROUTE_STOP id=”3" desc=”Graphical Pick: 2880,74686193883, 407235,024384864">
      <POINT x=”-95.9654113081327" y=”41.1342338099875"/>
     </ROUTE_STOP>
      <ROUTE_STOP id=”4" desc=”Graphical Pick: 117371,675366607, 165386,056416742">
      <POINT x=”-94.63261518695" y=”38.9680395256301"/>
     </ROUTE_STOP>
   </STOPSLIST>
   <ROUTE_PROPERTIES>
     <ROUTETYPE>quickest</ROUTETYPE>
     <PRECISION>0.0005</PRECISION>
         <SPEED_FACTOR>1.65</SPEED_FACTOR>
   </ROUTE_PROPERTIES>
  </GET_ROUTE>
</REQUEST>
</ARCXML>
```

START_TIME

Used in: REQUEST
Parent element: DRIVING_WINDOW

<START_TIME >
No attributes

No child elements

</START_TIME >

Description:

Defines a time when driving should start. Used for multiday routes.

Restrictions:

- The value of the element should be specified in the format hh:mm. Otherwise an error will be returned.
- END_TIME should be later than START_TIME.

Notes:

This value will be used as a time part of combined TRIP_START when explicit setting of TRIP_START element is omitted. See DRIVING_WINDOW for details.

Example for START_TIME:

```
<?xml version=”1.0" encoding=”UTF-8" ?><ARCXML version=”1.1">
<REQUEST>
<GET_ROUTE name=”1">
 <STOPSLIST>
  <ROUTE_STOP id=”1" desc=”Start”><POINT x=”-71.47288298" y=”41.786606252"/></ROUTE_STOP>
  <ROUTE_STOP id=”1" desc=”Stop1"><POINT x=”-71.43191514" y=”41.778305577"/></ROUTE_STOP>
  <ROUTE_STOP id=”2" desc=”Finish”><POINT x=”-71.45869150" y=”41.742425244"/></ROUTE_STOP>
 </STOPSLIST>
 <ROUTE_PROPERTIES>
  <TRIP_START>2006-09-26 17:55</TRIP_START>
  <DRIVING_DAY>
   <DRIVING_WINDOW>
    <START_TIME>09:00</START_TIME>
    <END_TIME>18:00</END_TIME>
   </DRIVING_WINDOW>
  </DRIVING_DAY>
  <ROUTETYPE>quickest</ROUTETYPE> <HWYPREF>50</HWYPREF> <PRECISION>0.00005</PRECISION>
 </ROUTE_PROPERTIES>
</GET_ROUTE>
</REQUEST>
</ARCXML>
```

STOPSLIST

Used in: REQUEST
Parent element: GET_ROUTE

<STOPSLIST >
No attributes

(m) <ROUTE_STOP... />

</STOPSLIST >

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Defines a list of routing stops.

Restrictions:

A minimum of two routing stops are required using ROUTE_STOP.

Notes:

None

Example for STOPSLIST:

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

```
        <ROUTE_STOP id="0" desc="75080">
          <POINT x="-96.743072" y="32.971216"/>
        </ROUTE_STOP>
        <ROUTE_STOP id="1" desc="75081">
          <POINT x="-96.710713" y="32.948611"/>
        </ROUTE_STOP>
      </STOPSLIST>
      <ROUTEFILTER>
        <ENVELOPE minx="-180.0063613231552" miny="11.358778625954183" maxx="-64.99363867684477"
maxy="78.64122137404581"/>
      </ROUTEFILTER>
      <ROUTE_PROPERTIES>
        <ROUTETYPE>quickest</ROUTETYPE>
        <HWYPREF>50</HWYPREF>
        <PRECISION>0.1463266191428886</PRECISION>
       <LOCALE language="en" country="US" variant="" />
       <LENGTH_UNITS name="kilometer"/>
      </ROUTE_PROPERTIES>
    </GET_ROUTE>
  </REQUEST>
</ARCXML>
```

TOTALS

Used in: RESPONSE
Parent element: ROUTE

```
<TOTALS
    distance =“double”
     time =“double”
>
    (m) <DESCRIPTION... />
 </TOTALS >
```

Bold: Attribute or child element is required.
(m): Child element can be used multiple times.

Description:

Contains summary information about total time and distance of route.

Restrictions:

None

Notes:

None

Attribute descriptions for TOTALS:

Attribute	Usage
distance	Total driving distance in miles.
time	Total driving time in minutes.

Example for TOTALS:

```
<ARCXML version="1.1">
<RESPONSE>
<ROUTE name="">
  <ENVELOPE minx="-118.366388" miny="34.164836" maxx="-118.332836"
maxy="34.175727"/>
  <PARTS>
   <PART fromstop="0" tostop="1" >
    <SEGMENT distance="0.000000" time="0.000000">
     <ENVELOPE minx="-118.366388" miny="34.175727" maxx="-118.366388" maxy="34.175727"/>
     <DESCRIPTION type="depart">Depart Lat: 34.17556482872054
Long:-118.36638757243382</DESCRIPTION>
      <POLYLINE>
       <PATH>
         <COORDS>-118.366388 34.175727;-118.366388 34.175727</COORDS>
        </PATH>
      </POLYLINE>
    </SEGMENT>
   </PART>
 </PARTS>
 <TOTALS distance="4.000668" time="8.433333">
    <DESCRIPTION type="totals">Driving distance: 4 mile(s)</DESCRIPTION>
    <DESCRIPTION type="totals">Driving time: 8 minute(s)</DESCRIPTION>
  </TOTALS>
</ROUTE>
</RESPONSE>
</ARCXML>
```

TRIP_START

Used in: REQUEST
Parent element: ROUTE_PROPERTIES

<TRIP_START>
No attributes
No elements
</TRIP_START >

Description:

Specifies local date and time when user departs from a route start point.

Restrictions:

- The value of the element should be specified in the format YYYY-MM-DD hh:mm; otherwise, an error will be returned.
- TRIP_START should be set if DRIVING_WINDOW is used.

Notes:

- Usage of TRIP_START results to ETA (estimated time of arrival) reporting for the each segment of resulted route.
- TRIP_START is local time (LT) of the departure. ETA reported will be also in departure local time.

Example for TRIP_START:

```
<?xml version=”1.0" encoding=”UTF-8" ?><ARCXML version=”1.1">
<REQUEST>
<GET_ROUTE name=”1">
 <STOPSLIST>
  <ROUTE_STOP id=”1" desc=”Start”><POINT x=”-71.47288298" y=”41.786606252"/></ROUTE_STOP>
  <ROUTE_STOP id=”1" desc=”Stop1"><POINT x=”-71.43191514" y=”41.778305577"/></ROUTE_STOP>
  <ROUTE_STOP id=”2" desc=”Finish”><POINT x=”-71.45869150" y=”41.742425244"/></ROUTE_STOP>
 </STOPSLIST>
 <ROUTE_PROPERTIES>
  <TRIP_START>2006-09-26 17:55</TRIP_START>
  <ROUTETYPE>quickest</ROUTETYPE> <HWYPREF>50</HWYPREF> <PRECISION>0.00005</PRECISION>
 </ROUTE_PROPERTIES>
</GET_ROUTE>
</REQUEST>
</ARCXML>
```

Glossary

address candidates

Potential matches for an address.

address matching

See Geocoding.

ArcIMS

ESRI software that allows Internet mapping and distributed GIS solutions. The administrative framework lets users author map configuration files, publish Services, design Web pages, and administer ArcIMS Spatial Servers. ArcIMS supports Windows® and UNIX® platforms and is customizable on many levels.

ArcIMS Administrator

The ArcIMS component that allows users to manage Services, Servers, and Virtual Servers.

ArcIMS Route Server

An ArcIMS extension that allows you to add routing, geocoding, and reverse geocoding of Smart Data Compression data to an ArcIMS Web site.

ArcIMS Spatial Server

The ArcIMS Spatial Server is the backbone of ArcIMS. It serves data to the client applications in the requested format. An ArcIMS Spatial Server's operation is defined by one or more Virtual Servers.

ArcIMS Virtual Server

A grouping of one or more ArcIMS Spatial Servers into one unit for administrative purposes.

ArcXML

The Arc Extensible Markup Language (ArcXML) file format provides a structured method for communication between all ArcIMS components. ArcXML defines content for Services and is used for requests and responses between clients, the business logic tier, and servers.

client

A computer and/or application that allows Internet users to communicate with a server.

drive time polygon

It is a zone of the map having each point of it accessible from the center point of the zone. Zone size is specified in either minutes or miles.

dynamic barrier

An obstruction on a road or street that prevents further routing along that segment. Barriers are stored in the SDE table located in the ArcSDE database. Route Server periodically polls this table to build a valid route.

geocoding

The mechanism that allows you to use addresses to identify locations on a map.

Geocoding style

Geocoding style is a set of address standardization and matching rules used for geocoding. These rules are country specific and may differ in the US, Canada and European countries.

Image Server

The Image Server is a public ArcIMS Virtual Server for Image Services.

map configuration file

Provides data layer content and symbology that the Service registers to the ArcIMS Spatial Server and Web server for processing.

restriction

Street segments can be restricted because of different reasons (pedestrian, carpool, minimum or maximum weight or height etc). In any case the restricted street segments will not be taken into account while performing of routing.

reverse geocoding

The process of finding an address by pointing at a location on the map.

RouteServer Virtual Server

The RouteServer Virtual Server is a private ArcIMS Virtual Server that supports routing on data prepared in Smart Data Compression (SDC) format. This server is available as an extension to ArcIMS.

routing

A path between two or more points. In ArcIMS Route Server, a route consists of a collection of segments, each containing descriptions, turn information, drive-time information, and segment distance, as well as the exact location of each segment. Locations are then used to plot the course on the map.

SDCGeocodeServer Virtual Server

The SDCGeocodeServer Virtual Server is a private ArcIMS Virtual Server that supports geocoding and reverse geocoding of data prepared in Smart Data Compression format. This server is available as an extension to ArcIMS.

Service

A Service allows the content of a map configuration file to be published on the Internet.

Smart Data Compression (SDC)

A highly compressed, indexed data format created specifically for optimizing routing and address matching in ArcIMS Route Server.

speed factor

This coefficient allows adjusting of speeds to calculate time in routes.

Stops order optimization

ArcIMS Route Server feature. Start and Finish stops are fixed, while intermediate stops are reordered to optimize the route. The optimization is possible when you have four or more stops (at least two intermediate stops are required).

Virtual Server

See ArcIMS Virtual Server.

XML

Extensible Markup Language. A World Wide Web standard used to create ArcXML.

Index

A

ADDRESS 40
Address 4
  candidate 179
  matching 1, 179
aimsrt.cfg 23, 32
  parameters 24
ArcIMS 179
ArcIMS Author 179
ArcIMS Knowledge Base 14
ArcIMS Route Server 1, 179
ArcIMS Spatial Server 179
ArcIMS Virtual Server 179
ARCXML 42
ArcXML 1, 179
  conventions 35
  schema 34

B

BARRIER_POINT 46
BARRIERLIST 44
Barriers 13, 23, 44
  Dynamic 23
  Per-request 23

C

CLEARRESTRICTION 48
Client 179
COORDS 50
Custom clients 1

D

DESCRIPTION 52
Driving directions 1
DRIVING_DAY 58
DRIVING_WINDOW 60
DTCENTER 62
DTPROPERTIES 66
DTTYPE 68
DTZONEINPUT 70
DTZONERESULT 72
DTZONES 74

E

END_TIME 76
ENVELOPE 78, 115
ERROR 82
ESRI 14
estimated time of arrival 176, 194
EXTENSION 31, 85

F

FEATURE 89
FEATURECOORDSYS 13, 20, 91
FIELD 93
FIELDVALUE 96
FILTERCOORDSYS 13, 20, 98

G

GCCOUNT 101
GCINPUT 103
GCPARAM 105
GCSTYLE 107
GCTAG 110
GEOCODE 113
Geocoding 25, 180
  parameters 27, 105
  reverse intersection 13, 28
  styles 13, 25, 26

GET_DRIVETIME 115
GET_GEOCODE 18, 117
GET_ROUTE 18, 119
GET_SERVICE_INFO 121

H

HWYPREF 122

I

Image Server 16, 180

K

Knowledge Base 14

L

LAYER 124
LAYERINFO 128
LENGTH_UNITS 130
LOCALE 30, 132

M

Map configuration file 180

O

OPTIMIZE_STOPS 13, 21, 22, 135

P

PART 137
PARTS 139
PATH 141
POINT 143, 156, 158, 160
POLYGON 146
POLYLINE 148
PRECISION 150

R

REQUEST 152
RESPONSE 154
RESTRICTION 156
RESTRICTIONS 158
RESTRICTIONSLIST 160
Reverse geocoding 4
ROUTE 162
Route Server 180
Route Virtual Server 16
  adding an additional 17
ROUTE_PROPERTIES 165
ROUTE_STOP 167
ROUTEFILTER 169
ROUTETYPE 171
Routing 1

S

SDC 31, 180
SDCWORKSPACE 173
SEGMENT 175
SERVICEINFO 180, 182, 185
Services 16
  advanced setting 21
  checking 17
  creating a route 22
SETRESTRICTION 182
Smart Data Compression (SDC) 180
SPEED_FACTOR 185
START_TIME 188
STOPSLIST 190
Street ID 23, 24

T

TOTALS 192, 194
TRIP_START 194

V

Virtual Server 16, 17, 18

X

XML 181