511 Regional Real-Time Transit System

Extensible Markup Language (XML) Document Type Definitions (DTDs) for Web Service Implementation

Version 1.9

(January 25, 2012)

Introduction

This document contains two sections. The first section includes the Extensible Markup Format (XML) document type definitions (DTDs) for the messages exchanged between the transit agency web services and the 511 system. The second section includes the Web Service Description Language (WSDL). All sections are based upon research conducted by the Metropolitan Transportation Commission (MTC) and the Telvent Farradyne Corporation.

Please note that transit agencies should only use this document if MTC has granted approval to use web services as the method to transfer real-time transit data to the regional real-time system. The preferred transfer method is using JMS. MTC will provide a sample JMS application with sample data to enable this transfer. However, if circumstances exist and MTC approves, transit agencies can build the data transfer using web services. This document lays out the web services definitions to which transit agencies must build and adhere.

Configuration Request Message

To obtain updated configuration information from the transit agencies, 511 issues a configuration request message. Each record may optionally contain a list of agency names. When an agency name(s) is specified, the configuration data contained in the response should be restricted to only the agencies specified. If no agency names are specified, all data will be returned. Note that while agencies would only return data for that specific agency, vendors that retain data for multiple agencies would return data for all agencies that meet the input criteria.

The format of the message is documented in the XML DTD shown below:

Configuration Request:

```
<!-- Request sent for on-demand updates for agency
     configuration information. -->
<!DOCTYPE SystemInformationRequest [
<!ELEMENT SystemInformationRequest>
<!ATTLIST SystemInformationRequest
      Agency      CDATA  #Format: Comma separated list>
 ]>
```

Configuration Response Message

Upon issuance of a configuration request message, a configuration response message will be sent. The response message will contain either a list of errors or the requested configuration data and optionally, a record containing a mapping between trip IDs and direction codes if necessary. If present, an error record will contain the following information (data element names are shown in parenthesis):

- The integer error code if applicable (**`errorCode`**)
- The text of the error (**`errorText`**)

Each trip mapping data record must contain the following:

- The name of the agency (**`agency`**)
- The version number of the mapping data (used to determine when changes have occurred since the last update) (**`version`**)
- The date/time of the current snapshot of data (**`TimeStamp`**)
- The list of trip ID mappings for the agency (**`TripMapping`**)

Each trip mapping record contains the following (some fields are optional):

- The identifier that distinguishes the trip stop along the route (**`tripID`**)
- An identifier or alias that uniquely distinguishes each route from all others (**`routeKey`**)
- An identifier or alias that uniquely distinguishes each direction from all others (**`dirKey`**)
- An optional flag (true/false) indicating if route/direction combination represented by the trip ID contains all stops (**`defaultTripForRouteDir`**)

Each configuration data record contains the following (some fields are optional):

- The name of the agency (**`agency`**)
- The version number of the configuration data (used to determine when changes have occurred since the last configuration update) (**`version`**)
- The date/time of the current snapshot of data (**`TimeStamp`**)
- The total number of stops (**`numStops`**)
- An optional description of the data – e.g. description of data quality (**`msg`**)
- The list of bus routes for the agency (**`Route`**)

Each agency may have one or more bus routes. Each bus route record may contain the following information (data element names are shown in parenthesis):

- An identifier or alias that uniquely distinguishes each route from all others (**`key`**)
- The name of the route – usually a street name (**`title`**)
- The list of directions and/or destinations of the route (**`Direction`**)

Each route may have one or more directions/destinations. Each direction/destination record contains the following information (data element names are shown in parenthesis - some fields are optional):

- An identifier or alias that uniquely distinguishes each direction from all others (**`key`**)

- The name of the direction – for example inbound or outbound (**`title`**)
- The type of the identifier – either DIRECTION_CODE or TRIP_ID (**`dirType`**)
- The list of stops in each direction/destination (**`Stop`**)
- An optional flag (true/false) indicating if direction contains all stops for the route (**`containsAllStopsInRoute`**)

Each direction may have one or more stops. Each stop record contains the following information (data element names are shown in parenthesis - some fields are optional):

- An identifier or alias that uniquely distinguishes each stop from all others (**`key`**)
- The name of the stop – usually the name of an intersection or station name (**`title`**)
- The position of the stop along the route (**`stopOrder`**)

The hierarchical relationship between all of the aforementioned data elements is documented in the XML DTD shown below.

`Configuration Response:`

```
<!-- Configuration information response received. -->
<!DOCTYPE ConfigurationDataMessage [
<!ELEMENT ConfigurationDataMessage (Error+ | (TripMappingData+,
ConfigurationData+) ) >
<!ELEMENT Error >
<!ATTLIST Error
      errorCode         CDATA #IMPLIED
      errorText         CDATA #REQUIRED >
<!ELEMENT TripMappingData (TripMapping+) >
<!ATTLIST TripMappingData
      Agency            CDATA #REQUIRED
      version           CDATA #REQUIRED
      TimeStamp         CDATA #REQUIRED
                              #Format: YYYY-MM-DDTHH:MM:SS.sssZ >
<!ELEMENT TripMapping >
<!ATTLIST TripMapping
      tripID            CDATA #REQUIRED
      routeKey          CDATA #REQUIRED
      dirKey            CDATA #REQUIRED
      defaultTripForRouteDir (TRUE | FALSE) >
<!ELEMENT ConfigurationData (Route+ ) >
<!ATTLIST ConfigurationData
      agency      CDATA #REQUIRED
      version           CDATA #REQUIRED
      numStops          CDATA #REQUIRED
      TimeStamp         CDATA #REQUIRED
                              #Format: YYYY-MM-DDTHH:MM:SS.sssZ
      msg               CDATA >
<!ELEMENT Route (Direction+) >
<!ATTLIST Route
      key         CDATA #REQUIRED
      title       CDATA #REQUIRED >
<!ELEMENT Direction (Stop+ ) >
<!ATTLIST Direction
      key         CDATA #REQUIRED
      title       CDATA #IMPLIED
      dirType     (DIRECTION_CODE | TRIP_ID) #REQUIRED
      containsAllStopsInRoute (TRUE | FALSE) >

<!ATTLIST Stop
      key         CDATA #REQUIRED
```

```
title       CDATA #REQUIRED
stopOrder   CDATA >]>
```

Prediction Request Message

To obtain updated prediction information from the agencies, 511 issues a prediction request message. Each record may optionally contain a list of agency names. When an agency name(s) is specified, the prediction data contained in the response should be restricted to only the agencies specified. If no agency names are specified, all data will be returned. Note that while agencies would only return data for that specific agency, vendors that retain data for multiple agencies would return data for all agencies. The format of the message is documented in the XML DTD shown below:

Prediction Request:

```
<!-- Request sent for on-demand updates for arrival time
     prediction information. -->
<!DOCTYPE PredictionRequest [
<!ELEMENT PredictionRequest >
<!ATTLIST PredictionRequest
      Agency      CDATA #IMPLIED #Format: Comma separated list>
 ]>
```

Prediction Data Message

To obtain departure/arrival time predictions, the 511 system (as well as any other applications) will issue a request. A prediction data message will be sent in response. The response message will contain either a list of errors or the requested prediction data. An error record will contain the following information (data element names are shown in parenthesis):

- The integer error code if applicable (**`errorCode`**)
- The text of the error (**`errorText`**)

Each prediction data record must contain the following information (data element names are shown in parenthesis):

- The name of the agency (**`agency`**)
- The version number of the configuration data (used to determine when changes have been since the last configuration update) (**`version`**)
- The date/time of the current snapshot of data (**`TimeStamp`**)
- The total number of stops (**`numStops`**)
- The list of departure/arrival time predictions – one for each route (**`StopPredictions`**)

Each agency may have one or more bus routes. Each bus route may possess multiple directions/destinations and may have several stops. Thus, each route must contain a list of the departure/arrival time predictions for each stop in <u>all</u> directions/destinations. Each prediction record must contain the following information (data element names are shown in parenthesis):

- The identifier or alias that uniquely distinguishes each route (**`route`**)
- The name of the direction/destination (**`dir`**)
- The identifier or alias that uniquely distinguishes each stop (**`stop`**)
- An optional message containing information concerning the route (**`msg`**)
- A list of predicted arrival and departure times for each stop (**`Ptimes`**)
- The type of prediction; either arrival or departure (**`PredictionType`**)
- Each stop may have one or more departure/arrival time predictions. Each departure/arrival time record must contain the time at which a vehicle is predicted to arrive at or depart from the stop (**`PredictionTime`**).
- Each departure/arrival time prediction is associated with a trip (tripID)

Each agency may have one or more bus routes. Each bus route may possess multiple directions/destinations and may operate several trips. Each trip record for a particular route and direction may contain the following information (data element names are shown in parenthesis):

- Each trip record will contain optional trip id (tripID)
- Each trip record will contain optional vehicle location information for that trip (vehicleLat, vehicleLong)

The hierarchical relationship between all of the aforementioned data elements is documented in the XML DTD shown below.

Prediction Response:

```
<!-- Prediction information received. -->
<!DOCTYPE PredictionDataMessage [
<!ELEMENT PredictionDataMessage (Error* | PredictionData*) >
<!ELEMENT Error >
<!ATTLIST Error
      errorCode         CDATA #IMPLIED
      errorText         CDATA #REQUIRED >
<!ELEMENT PredictionData (Predictions+, VehicleLocationData*) >
<!ATTLIST PredictionData
      agency      CDATA #REQUIRED
      version           CDATA #REQUIRED
      numStops          CDATA #REQUIRED
      TimeStamp         CDATA #REQUIRED
                              #Format: YYYY-MM-DDTHH:MM:SS.sssZ >
<!ELEMENT StopPredictions (Ptimes*) >
<!ATTLIST StopPredictions
      route             CDATA #REQUIRED
      dir               CDATA #REQUIRED
      stop              CDATA #REQUIRED
      msg               CDATA #IMPLIED >
<!ATTLIST Ptimes

      PredictionType     (D | A ) #REQUIRED
                              #Format: D - Departure Time
                              #        A - Arrival Time

      PredictionTime    CDATA #REQUIRED

                              #Format: YYYY-MM-DDTHH:MM:SS.sssZ

      tripID            CDATA #IMPLIED >

<!ELEMENT VehicleLocationData (VehicleLocation*) >

<!ATTLIST VehicleLocationData

      route        CDATA #REQUIRED

      dir          CDATA #REQUIRED  >

<!ELEMENT VehicleLocation EMPTY  >

<!ATTLIST VehicleLocation

      tripID            CDATA #IMPLIED

      vehicleLat        CDATA #IMPLIED

      vehicleLong       CDATA #IMPLIED>  ]>
```

Arrived Status Request Message

To obtain updated actual arrival times of the vehicles from the agencies, 511 issues an arrived status request message to the web server that services the specific agency. Each arrived status request record will contain the following attributes (data element names are shown in parenthesis). Note: for greater system flexibility, the request message is being designed to allow for both specific information requests (e.g. arrived status at one or more stops) and general information requests (e.g. arrived status for all stops):

- The name of the agency (**`agency`**) – may be a single entry or a comma-separated list
- The identifier or alias that uniquely distinguishes each route (**`route`**) – may be a single entry or a comma-separated list
- The identifier of the direction/destination (**`dir`**) – may be a single entry or a comma-separated list
- The identifier of the stop (**`stop)`** – may be a single entry or a comma-separated list
- The start time for which data is requested (**`StartTime)`**
- The stop time for which data is requested (**`StopTime)`**
- The identifier of the trip (**`TripId)-`** may be a single entry or a comma-separated list

In the request message all of the attributes are flexible. Flexible attributes mean that the request might include a specific attribute (e.g. list of specific stop codes) or an attribute that is left blank or not included. If an attribute is left blank or not included, all data for that attribute will be retrieved (e.g. if attribute 'route' is null or blank, the reply message will include data for all routes). The more attributes that are left blank or not included will result in more information being returned. Because of this, it is expected that a list of stops for which arrival times are requested will generally be provided.

The starttime and stoptimes fields are the time range for which status is being requested. Typically, this message should be requested once per day, and should be for the full previous day. For example, the Arrived Status Request Message could be configured to be sent once per day at 2AM. If it was sent on February 2, 2006, the start date would be 2/1/2006 12:00:00 AM, and end date would be 2/1/2006 11:59:59 PM.

The format of the message is documented in the XML DTD shown below:

Arrival Status Request:

```
<!-- Request sent to the agencies for updates for actual vehicle arrival and
departure times. -->
<!DOCTYPE ArrivalStatusRequest [
<!ELEMENT ArrivalStatusRequest>
<!ATTLIST ArrivalStatusRequest
      agency            CDATA #IMPLIED #Format: Comma separated list
      route             CDATA #IMPLIED #Format: Comma separated list
      dir               CDATA #IMPLIED #Format: Comma separated list
      stop              CDATA #IMPLIED #Format: Comma separated list
      startTime         CDATA #IMPLIED #Format: YYYY-MM-DDTHH:MM:SS.sssZ
      stopTime          CDATA #IMPLIED #Format: YYYY-MM-DDTHH:MM:SS.sssZ
      tripId      CDATA #IMPLIED #Format: Comma separated list>
]>
```

Arrived Status Response Message

In response to arrived status request messages, an arrived status response message will be sent. The response message will contain either a list of errors or the requested arrival status data. If present, an error record will contain the following information (data element names are shown in parenthesis):

- The integer error code if applicable (**`errorCode`**)
- The text of the error (**`errorText`**)

Each arrived status response record must contain the following information (data element names are shown in parenthesis):

- The name of the agency (**`agency`**)
- The date/time of the current snapshot of data (**`TimeStamp`**)
- The identifier or alias that uniquely distinguishes each route (**`route`**)
- The name of the direction/destination (**`dir`**)
- The identifier or alias that uniquely distinguishes each stop (**`stop`**)
- The unique identifier of the trip (**`TripId`**)
- The time at which the bus arrived at the stop (**`ArrivalTime`**)

The relationship between all of the aforementioned data elements is documented in the XML DTD shown below.

`Arrival Status Response:`

```
<!-- Arrival Status information received. -->
<!DOCTYPE ArrivalStatusDataMessage [
<!ELEMENT ArrivalStatusDataMessage (Error* | ArrivalStatusDataMessage*) >
<!ELEMENT Error >
<!ATTLIST Error
      errorCode         CDATA #IMPLIED
      errorText         CDATA #REQUIRED >
<!ELEMENT ArrivalStatusData >
<!ATTLIST ArrivalStatusData
      agency      CDATA #REQUIRED
      TimeStamp         CDATA #REQUIRED
                              #Format: YYYY-MM-DDTHH:MM:SS.sssZ
      route             CDATA #REQUIRED
      dir               CDATA #REQUIRED
      stop              CDATA #REQUIRED
      TripId      CDATA #REQUIRED
      ArrivalTime       CDATA #REQUIRED
                              #Format: YYYY-MM-DDTHH:MM:SS.sssZ >
]>
```

Web Service Description Language (WSDL)

Below is a Web Service Description Language (WSDL) document describing the methods available to web clients to request and receive configuration, prediction, and arrival data from transit agencies.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
File Name     :  AgencyDataService.wsdl
Prepared by   :  Telvent/Farradyne

   Copyright 2007, Telvent/Farradyne Inc.
                   All Rights Reserved

This file is property of Metropolitan Transportation Commission (MTC).  Any unauthorized use or
duplication of this file, or removal of the header, constitutes
theft of intellectual property.

Description   :  This file contains the WSDL for agency data for the 511 project.  Descriptions
of
                 the three (3) public web methods follows:

   getAgencyConfigurationData - Request for agency configuration data.
        parameter in0 - Optional comma separated list of agency names for which configuration
data will be returned.
        returns - ConfigurationDataMessage containing either an array of ConfigurationData
objects or an array
                  of Error objects.

   getPredictionData - Request for vehicle arrival prediction updates.
        parameter in0 - Optional comma separated list of agency names for which prediction data
will be returned.
        returns - PredictionDataMessage containing either an array of PredictionData objects or
an array
                  of Error objects.

   getArrivalStatusData - Request for actual vehicle arrival/departure updates.
        parameter in0 - Optional comma separated list of agency names for which arrival status
data will be returned.
        parameter in1 - Optional comma separated list of route codes for which arrival status
data will be returned.
        parameter in2 - Optional comma separated list of direction/destination codes for which
arrival status data will be returned.
        parameter in3 - Optional comma separated list of stop codes for which arrival status data
will be returned.
        parameter in4 - Start time. If specified, only arrival times after the start time will be
returned.
        parameter in5 - Stop time. If specified, only arrival times before the start time will be
returned.
```

```
        returns - ArrivalStatusDataMessage containing either an array of ArrivalStatusData
objects or an array

                  of Error objects.

History:

-->

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

<wsdl:definitions targetNamespace="urn:AgencyDataServiceService"
xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:impl="urn:AgencyDataServiceService"
xmlns:intf="urn:AgencyDataServiceService" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<!--WSDL created by Apache Axis version: 1.4

Built on Apr 22, 2006 (06:55:48 PDT)-->

 <wsdl:types>

  <schema elementFormDefault="qualified" targetNamespace="urn:AgencyDataServiceService"
xmlns="http://www.w3.org/2001/XMLSchema">

   <element name="getArrivalStatusData">

    <complexType>

     <sequence>

      <element name="agency" type="xsd:string"/>

      <element name="route" type="xsd:string"/>

      <element name="dir" type="xsd:string"/>

      <element name="stop" type="xsd:string"/>

      <element name="startTime" type="xsd:dateTime"/>

      <element name="stopTime" type="xsd:dateTime"/>

     </sequence>

    </complexType>

   </element>

   <element name="getArrivalStatusDataResponse">

    <complexType>

     <sequence>

      <element name="getArrivalStatusDataReturn" type="impl:ArrivalStatusDataMessage"/>

     </sequence>

    </complexType>

   </element>

   <complexType name="ArrivalStatusData">

    <sequence>

     <element name="agency" nillable="true" type="xsd:string"/>

     <element name="arrivalTime" nillable="true" type="xsd:dateTime"/>

     <element name="dir" nillable="true" type="xsd:string"/>

     <element name="route" nillable="true" type="xsd:string"/>

     <element name="stop" nillable="true" type="xsd:string"/>

     <element name="tripID" nillable="true" type="xsd:string"/>

    </sequence>

   </complexType>
```

```
<complexType name="ArrayOfArrivalStatusData">
 <sequence>
  <element maxOccurs="unbounded" minOccurs="0" name="item" type="impl:ArrivalStatusData"/>
 </sequence>
</complexType>
<complexType name="Error">
 <sequence>
  <element name="errorCode" type="xsd:int"/>
  <element name="errorText" nillable="true" type="xsd:string"/>
 </sequence>
</complexType>
<complexType name="ArrayOfError">
 <sequence>
  <element maxOccurs="unbounded" minOccurs="0" name="item" type="impl:Error"/>
 </sequence>
</complexType>
<complexType name="ArrivalStatusDataMessage">
 <sequence>
  <element name="arrivalStatus" nillable="true" type="impl:ArrayOfArrivalStatusData"/>
  <element name="errors" nillable="true" type="impl:ArrayOfError"/>
 </sequence>
</complexType>
<element name="getPredictionData">
 <complexType>
  <sequence>
   <element name="agency" type="xsd:string"/>
  </sequence>
 </complexType>
</element>
<element name="getPredictionDataResponse">
 <complexType>
  <sequence>
   <element name="getPredictionDataReturn" type="impl:PredictionDataMessage"/>
  </sequence>
 </complexType>
</element>
<complexType name=" VehicleLocation ">
  <sequence>
   <element name="tripID" nillable="true" type="xsd:string"/>
   <element name="vehicleLat" nillable="true" type="xsd:double"/>
   <element name="vehicleLong" nillable="true" type="xsd:double"/>
  </sequence>
 </complexType>
 <complexType name="ArrayOfVehicleLocation">
  <sequence>
```

```
     <element maxOccurs="unbounded" minOccurs="0" name="item"
type="impl:VehicleLocation"/>
    </sequence>
   </complexType>
   <complexType name=" VehicleLocationData">
    <sequence>
     <element name="dir" nillable="true" type="xsd:string"/>
     <element name="route" nillable="true" type="xsd:string"/>
     <element name=" VehicleLocation" nillable="true"
type="impl:ArrayOfVehicleLocation"/>
    </sequence>
   </complexType>
   <complexType name="ArrayOfVehicleLocationData">
    <sequence>
     <element maxOccurs="unbounded" minOccurs="0" name="item"
type="impl:VehicleLocationData"/>
    </sequence>
   </complexType>   <complexType name="Ptimes">
    <sequence>
     <element name="predictionTime" nillable="true" type="xsd:dateTime"/>
     <element name="predictionType" nillable="true" type="xsd:string"/>
     <element name="tripId" nillable="true" type="xsd:string"/>    </sequence>
   </complexType>
   <complexType name="ArrayOfPtimes">
    <sequence>
     <element maxOccurs="unbounded" minOccurs="0" name="item" type="impl:Ptimes"/>
    </sequence>
   </complexType>
   <complexType name="StopPredictions">
    <sequence>
     <element name="dir" nillable="true" type="xsd:string"/>
     <element name="msg" nillable="true" type="xsd:string"/>
     <element name="ptimes" nillable="true" type="impl:ArrayOfPtimes"/>
     <element name="route" nillable="true" type="xsd:string"/>
     <element name="stop" nillable="true" type="xsd:string"/>
    </sequence>
   </complexType>
   <complexType name="ArrayOfStopPredictions">
    <sequence>
     <element maxOccurs="unbounded" minOccurs="0" name="item" type="implStopPredictions"/>
    </sequence>
   </complexType>
   <complexType name="PredictionData">
    <sequence>
     <element name="agency" nillable="true" type="xsdstring"/>
     <element name="numStops" type="xsdint"/>
```

```
  <element name="stopPredictions" nillable="true" type="impl:ArrayOfStopPredictions"/>
     <element name=" VehicleLocationData" nillable="true"
     type="impl:ArrayOfVehicleLocationData"/>
  <element name="timeStamp" nillable="true" type="xsddateTime"/>
  <element name="version" nillable="true" type="xsdstring"/>
 </sequence>
</complexType>
<complexType name="ArrayOfPredictionData">
 <sequence>
  <element maxOccurs="unbounded" minOccurs="0" name="item" type="impl:PredictionData"/>
 </sequence>
</complexType>
<complexType name="PredictionDataMessage">
 <sequence>
  <element name="errors" nillable="true" type="impl:ArrayOfError"/>
  <element name="predictions" nillable="true" type="impl:ArrayOfPredictionData"/>
 </sequence>
</complexType>
<element name="getAgencyConfigurationData">
 <complexType>
  <sequence>
   <element name="agency" type="xsd:string"/>
  </sequence>
 </complexType>
</element>
<element name="getAgencyConfigurationDataResponse">
 <complexType>
  <sequence>
   <element name="getAgencyConfigurationDataReturn" type="impl:ConfigurationDataMessage"/>
  </sequence>
 </complexType>
</element>
<complexType name="Stop">
 <sequence>
  <element name="key" nillable="true" type="xsd:string"/>
  <element name="position" nillable="true" type="xsd:string"/>
  <element name="stopOrder" type="xsd:int"/>
  <element name="title" nillable="true" type="xsd:string"/>
 </sequence>
</complexType>
<complexType name="ArrayOfStop">
 <sequence>
  <element maxOccurs="unbounded" minOccurs="0" name="item" type="impl:Stop"/>
 </sequence>
```

```
</complexType>
<complexType name="Direction">
 <sequence>
  <element name="containsAllStopsInRoute" type="xsd:boolean"/>
  <element name="dirType" nillable="true" type="xsd:string"/>
  <element name="key" nillable="true" type="xsd:string"/>
  <element name="stop" nillable="true" type="impl:ArrayOfStop"/>
  <element name="title" nillable="true" type="xsd:string"/>
 </sequence>
</complexType>
<complexType name="ArrayOfDirection">
 <sequence>
  <element maxOccurs="unbounded" minOccurs="0" name="item" type="impl:Direction"/>
 </sequence>
</complexType>
<complexType name="Route">
 <sequence>
  <element name="direction" nillable="true" type="impl:ArrayOfDirection"/>
  <element name="key" nillable="true" type="xsd:string"/>
  <element name="title" nillable="true" type="xsd:string"/>
 </sequence>
</complexType>
<complexType name="ArrayOfRoute">
 <sequence>
  <element maxOccurs="unbounded" minOccurs="0" name="item" type="impl:Route"/>
 </sequence>
</complexType>
<complexType name="ConfigurationData">
 <sequence>
  <element name="agency" nillable="true" type="xsd:string"/>
  <element name="msg" nillable="true" type="xsd:string"/>
  <element name="numStops" type="xsd:int"/>
  <element name="route" nillable="true" type="impl:ArrayOfRoute"/>
  <element name="timeStamp" nillable="true" type="xsd:dateTime"/>
  <element name="version" nillable="true" type="xsd:string"/>
 </sequence>
</complexType>
<complexType name="ArrayOfConfigurationData">
 <sequence>
  <element maxOccurs="unbounded" minOccurs="0" name="item" type="impl:ConfigurationData"/>
 </sequence>
</complexType>
<complexType name="TripMapping">
```

```
    <sequence>
     <element name="defaultTripForRouteDir" type="xsd:boolean"/>
     <element name="directionCode" nillable="true" type="xsd:string"/>
     <element name="routeCode" nillable="true" type="xsd:string"/>
     <element name="tripId" nillable="true" type="xsd:string"/>
    </sequence>
   </complexType>
   <complexType name="ArrayOfTripMapping">
    <sequence>
     <element maxOccurs="unbounded" minOccurs="0" name="item" type="impl:TripMapping"/>
    </sequence>
   </complexType>
   <complexType name="TripMappingData">
    <sequence>
     <element name="agency" nillable="true" type="xsd:string"/>
     <element name="timeStamp" nillable="true" type="xsd:dateTime"/>
     <element name="tripMappings" nillable="true" type="impl:ArrayOfTripMapping"/>
     <element name="version" nillable="true" type="xsd:string"/>
    </sequence>
   </complexType>
   <complexType name="ArrayOfTripMappingData">
    <sequence>
     <element maxOccurs="unbounded" minOccurs="0" name="item" type="impl:TripMappingData"/>
    </sequence>
   </complexType>
   <complexType name="ConfigurationDataMessage">
    <sequence>
     <element name="configurations" nillable="true" type="impl:ArrayOfConfigurationData"/>
     <element name="errors" nillable="true" type="impl:ArrayOfError"/>
     <element name="tripMappings" nillable="true" type="impl:ArrayOfTripMappingData"/>
    </sequence>
   </complexType>
  </schema>
 </wsdl:types>
   <wsdl:message name="getAgencyConfigurationDataResponse">
      <wsdl:part element="impl:getAgencyConfigurationDataResponse" name="parameters"/>
   </wsdl:message>
   <wsdl:message name="getPredictionDataRequest">
      <wsdl:part element="impl:getPredictionData" name="parameters"/>
   </wsdl:message>
   <wsdl:message name="getArrivalStatusDataRequest">
      <wsdl:part element="impl:getArrivalStatusData" name="parameters"/>
   </wsdl:message>
```

```
   <wsdl:message name="getPredictionDataResponse">
      <wsdl:part element="impl:getPredictionDataResponse" name="parameters"/>
   </wsdl:message>
   <wsdl:message name="getArrivalStatusDataResponse">
      <wsdl:part element="impl:getArrivalStatusDataResponse" name="parameters"/>
   </wsdl:message>
   <wsdl:message name="getAgencyConfigurationDataRequest">
      <wsdl:part element="impl:getAgencyConfigurationData" name="parameters"/>
   </wsdl:message>
   <wsdl:portType name="AgencyData">
      <wsdl:operation name="getArrivalStatusData">
         <wsdl:input message="impl:getArrivalStatusDataRequest"
name="getArrivalStatusDataRequest"/>
         <wsdl:output message="impl:getArrivalStatusDataResponse"
name="getArrivalStatusDataResponse"/>
      </wsdl:operation>
      <wsdl:operation name="getPredictionData">
         <wsdl:input message="impl:getPredictionDataRequest" name="getPredictionDataRequest"/>
         <wsdl:output message="impl:getPredictionDataResponse" name="getPredictionDataResponse"/>
      </wsdl:operation>
      <wsdl:operation name="getAgencyConfigurationData">
         <wsdl:input message="impl:getAgencyConfigurationDataRequest"
name="getAgencyConfigurationDataRequest"/>
         <wsdl:output message="impl:getAgencyConfigurationDataResponse"
name="getAgencyConfigurationDataResponse"/>
      </wsdl:operation>
   </wsdl:portType>
   <wsdl:binding name="AgencyDataServiceSoapBinding" type="impl:AgencyData">
      <wsdlsoap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsdl:operation name="getArrivalStatusData">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="getArrivalStatusDataRequest">
            <wsdlsoap:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="getArrivalStatusDataResponse">
            <wsdlsoap:body use="literal"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="getPredictionData">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="getPredictionDataRequest">
            <wsdlsoap:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="getPredictionDataResponse">
            <wsdlsoap:body use="literal"/>
```

```
            </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="getAgencyConfigurationData">
            <wsdlsoap:operation soapAction=""/>
            <wsdl:input name="getAgencyConfigurationDataRequest">
               <wsdlsoap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="getAgencyConfigurationDataResponse">
               <wsdlsoap:body use="literal"/>
            </wsdl:output>
         </wsdl:operation>
      </wsdl:binding>
      <wsdl:service name="AgencyDataService">
         <wsdl:port binding="impl:AgencyDataServiceSoapBinding" name="AgencyDataService">
            <wsdlsoap:address location="http://localhost:8080/axis/services/AgencyDataService"/>
         </wsdl:port>
      </wsdl:service>
   </wsdl:definitions>
```