

February 11th, 2011

San Gabriel Valley Pilot Project

INTERNET XML DATA EXPORT INTERFACE
(IEN OUTBOUND WEB SERVICE)

Deliverable 4.3 – Interface Design
Document (IDD)

Final, Revision 1

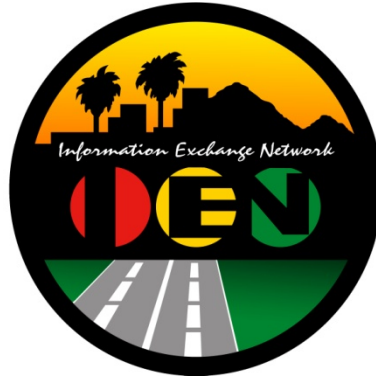


Prepared by:



626 Wilshire Blvd. Suite 818
Los Angeles, CA 90017

**LOS ANGELES COUNTY
INFORMATION EXCHANGE NETWORK**



**IEN XML DATA EXPORT INTERFACE
(IEN OUTBOUND WEB SERVICE)**

DELIVERABLE 4.3 – INTERFACE DESIGN DOCUMENT

FINAL, Revision 1

Prepared for:

**Los Angeles County
Department of Public Works**

Prepared by:

TRANSCORE

626 Wilshire Blvd.
Suite 818
Los Angeles, California 90017

February 11th, 2011

TABLE OF CONTENTS	PAGE #
1. INTRODUCTION	1-1
1.1 Purpose.....	1-1
1.2 Audience	1-1
1.3 References.....	1-1
1.4 Design Assumptions	1-2
1.5 Definitions, Acronyms, and Abbreviations	1-2
2. INTERFACE ARCHITECTURE.....	2-1
3. DATA CHARACTERISTICS.....	3-1
3.1 Data Update Rates.....	3-1
3.2 Data Availability	3-1
3.3 Data Latencies.....	3-2
4. INTERFACE OPERATIONS.....	4-1
4.1 OP_Register	4-1
4.2 Inputs.....	4-2
4.3 Outputs.....	4-2
4.4 Faults.....	4-4
4.5 OP_GetDeviceUpdate.....	4-5
4.6 Inputs.....	4-5
4.7 Outputs.....	4-7
4.7.1 IENSignalInventory.....	4-8
4.7.2 IENDetectorInventory	4-9
4.7.3 IENSignalSummary.....	4-9
4.7.4 IENDetectorSummary	4-9
4.7.5 IENLastCyclePhases	4-10
4.7.6 IENPlannedPhases.....	4-10
4.8 Faults.....	4-10
4.9 OP_UnRegister	4-10
4.10 Inputs.....	4-11
4.11 Outputs.....	4-12
4.12 Faults.....	4-12
5. APPENDICES	5-1
5.1 Appendix A: IEN Outbound Web Service WSDL	5-1
5.2 Appendix B: IEN Outbound Web Service XML Schema	5-4
5.3 Appendix C: Example IEN Outbound Web Service Requests and Responses	5-11
5.3.1 OP_Register.....	5-11
5.3.2 OP_GetDeviceUpdate	5-12
5.3.3 OP_UnRegister.....	5-21
5.4 Appendix D: Location of Required Data in the XML Schema	5-22
5.5 Appendix E: Traffic Control System Support for IEN data	5-25

REVISION HISTORY

Version	Date	Description	Reviewed By
Initial Draft	3/29/2010	First draft	Chuck Dankocsik
Final	8/16/10	Addresses draft comments and additional detector data requested by RIITS at the Walk-Thru Meeting (conducted 5/19/10).	Chuck Dankocsik
Final, Revision 1	2/11/11	Addresses additional LA County comments on Section 3.2 and adds Appendix E (submitted 9/14/10).	Chuck Dankocsik

1. INTRODUCTION

1.1 PURPOSE

This document describes the architecture, operations, and data set of the Los Angeles County Information Exchange Network (IEN) Outbound Web Service. In addition, it describes/documents the WSDL, XML schema, web service interface, and services the interface will expose to RIITS to meet the requirements and goals identified in the System Requirements Document (SRD).

This document is a deliverable of the IEN XML Data Export Interface Project.

1.2 AUDIENCE

This document is intended for the software developers who are interested in writing client applications for the IEN Outbound Web Service.

Readers of this document should be familiar with the following technologies:

- TCP/IP networking
- Web Services
- XML
- WSDL
- SOAP

1.3 REFERENCES

This document references the following documents:

- IEN XML Data Export Interface SOW, January 11th, 2010.
- IEN XML Data Export Interface System Requirements Document, January 25th, 2010.
- “Desired IEN Data for RIITS” document (dated 11/3/08).
- NTCIP 2306 v01.69r National Transportation Communications for ITS Protocol (NTCIP) Application Profile for XML Message Encoding and Transport in ITS Center-To-Center Communications (NTCIP C2C XML) , December, 2008.
- Traffic Management Data Dictionary (TMDD) V3.0, November 2008.
- Web Services Description Language (WSDL) Version 2.0 Part 0: Primer, <http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626/>
- Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, <http://www.w3.org/TR/wsdl20/>
- SOAP Version 1.2 specification, <http://www.w3.org/TR/soap12/>

1.4 DESIGN ASSUMPTIONS

The IEN Outbound Web Service design is based on the following assumptions:

- The IEN Outbound Web Service must export the IEN data identified in the “Desired IEN Data for RIITS” document (dated 11/3/08). The provision of additional IEN data sets would be supported under a future project and per LA County’s direction.
- LA County will use their firewall to control access to the IEN Web Server from external networks. Security measures such as data encryption and integrity checking are not required.
- External systems will connect to the IEN Web Server over private inter-Agency network links (e.g. leased lines, fiber connections, etc.). Support for connections over the Internet is not required.
- LA County will be responsible for securing permission from all IEN Agencies for the publication of their traffic data to external systems.
- The IEN Outbound Web Service will not export data that the IEN acquires from LADOT, RIITS, or PeMS (California Highway Performance Measurement System).
- LA County will control which member Agencies’ traffic data is available through the IEN Outbound Web Service.
- The initial version of the IEN Outbound Web Service will make the same set of data available to all web clients. The IEN Outbound Web Service will not need to restrict data based on the requesting external system. For example, the IEN Outbound Web Service will not need to be capable of allowing RIITS to collect Pasadena data, while denying Cal-Poly Pomona from collecting that same data.
- The IEN Outbound Web Service will run on a server computer located at the LA County Department of Public Works Transportation Management Center (LACO TMC).
- The IEN Web Server will be connected to the IEN Regional Server and IEN Database Server over a local area network with a bandwidth of 100Mbps (or better).

1.5 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

The following terms are used within this document:

Table 1-1: Definitions of Terms

TERM	DEFINITION
Arterial system detectors	A device used to measure traffic speed, volume, and occupancy across a particular section of roadway at an intersection.
Intersection controller	A device that controls the operation of an intersection’s vehicle and pedestrian signals. It can function independently or as part of a computerized traffic control system.
ITS	Intelligent Transportation Systems. Computerized systems for managing vehicular traffic.

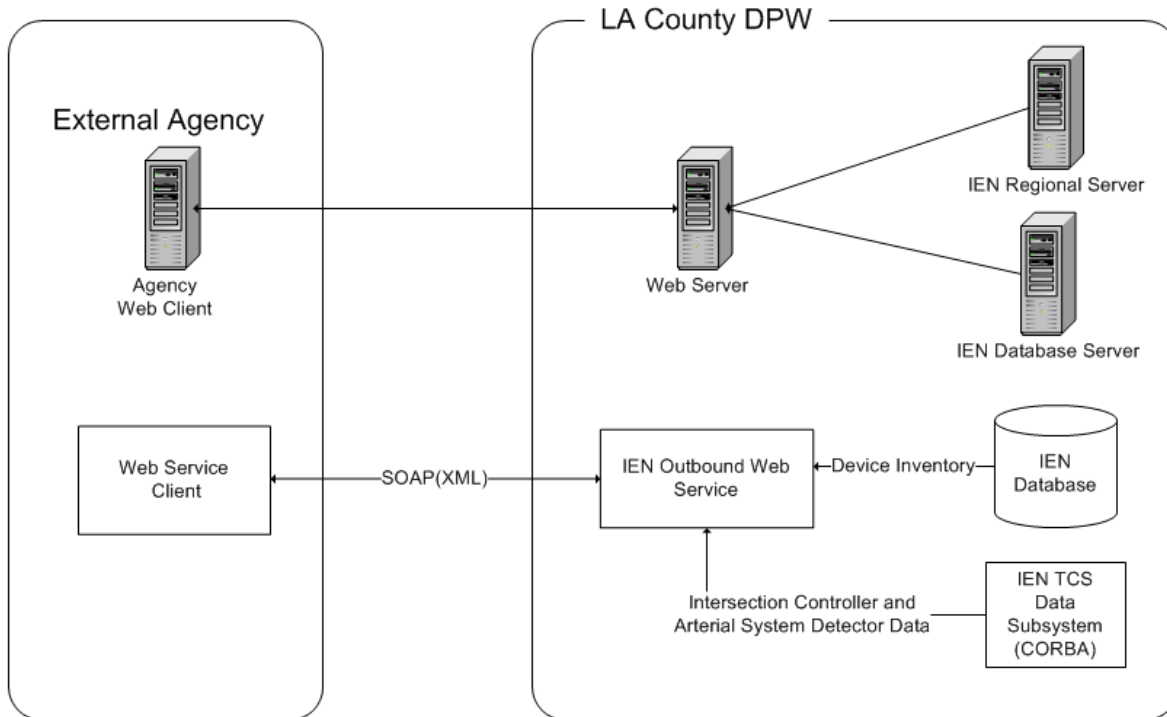
TERM	DEFINITION
NTCIP	National Transportation Communication for ITS Protocols. US national standard protocols for use with intelligent transportation systems.
SOAP	A protocol specification for exchanging structured information in XML formatted messages in the implementation of Web Services.
Signal	See intersection controller
TMDD	See Traffic Management Data Dictionary
Traffic Management Data Dictionary	Standard NTCIP dictionary of traffic management data items and messages. Usually referred to as the TMDD.
Volume	Vehicles per hour crossing a detector, link, or node.
Web Services	A software communication standard that supports data exchange between remote computers using web servers and data encoded in XML
Web Services Description Language	A standard XML schema for describing web services supported by a system.
WSDL	See Web Services Description Language
XML	Extensible Markup Language. A standard for representing structured data using text characters in a format similar to that used for web pages.
XML Schema	A document used to describe the XML messages that are sent via the services in the WSDL.
XSD	A standard XML schema document used to describe the XML messages that are sent via the services in the WSDL.

2. INTERFACE ARCHITECTURE

The IEN Outbound Web Service is a request/response-style Web Service interface that provides IEN intersection controller and arterial system detector data to external Agencies. Data requests and responses are exchanged as SOAP-encoded XML messages.

The high-level interface architecture is illustrated in Exhibit 2.1 below.

Exhibit 2.1: High-Level Interface Architecture



External web clients request data from the IEN web server. The web server retrieves data from the IEN's internal TCS data subsystem and sends that data back to requesting external Agencies.

The IEN Outbound Web Service WSDL is listed in Appendix A. The XML schema is listed in Appendix B.

3. DATA CHARACTERISTICS

Characteristics of IEN data are described in the following sections.

3.1 DATA UPDATE RATES

The IEN continuously collects intersection controller and arterial system detector data from connected traffic control systems. The IEN polls for a full set of summary status data for each device once-per-minute and for status changes the other 59 seconds of each minute. The IEN does full data polls for 1/60 of the total devices each second so as to not overwhelm traffic control systems with data requests. Intersection controller and arterial system detector data change at different rates; including cyclically (example: cycle length, volume), a few times per day (example: timing plan, control mode), and rarely (example: latitude).

The IEN Outbound Web Service reports the latest status data that is known to the IEN at the time that a web client request is received. It is expected that web clients will request data no more than once or twice per minute. The Web Server will not enforce a maximum request rate (at least initially). High request rates will, however, negatively impact the performance of the web server and so it is in the web client's best interest to limit the number requests they make per minute.

3.2 DATA AVAILABILITY

The IEN does not receive all required data for each device from member Agencies. The availability of data depends on the combined status and capabilities of the device, traffic control system, field-to-central communications, and command/data interface. For example, Agencies that use McCain QuicNet systems only report actual offset and planned phase max green times for controllers that run Bi Trans 233RV2 firmware and communicate at a minimum of 9600 baud rate.

Agency traffic control systems and the IEN data they support are listed in Appendix E. It is important to note that while a traffic control system may be capable of supporting a particular type of IEN data, that data may not be available for a given device because it has not been configured within the traffic control system. For example, detector direction and road name are usually optional fields within traffic control systems that many Agencies have not entered and therefore are not available to the IEN. The IEN Outbound Web Service handles requests for unavailable data in the following manner:

- If a particular piece of data (such as occupancy or status) is unavailable for a device, the field within the web service response will either be null or “-1” to indicate that the data is unknown.
- If the IEN is not receiving data from a member Agency, the web service will respond with a warning message that indicates that data is currently unavailable for that Agency.
- Due to the distributed nature of the IEN, it is not uncommon for network connections to occasionally bounce up and down. When a member Agency stops reporting data, the web service will continue to report the last received data from that Agency for several minutes until it clears that data for being too stale. A timestamp is included in IEN Outbound Web Service responses in the `last_update` element of the

ienDeviceUpdate structure (See Table 4-4). The timestamp represents the time at which the IEN last received an update for the device. External systems can use that timestamp to evaluate if the data is too stale for their purposes.

3.3 DATA LATENCIES

The IEN, the participating traffic control systems, and the Internet are interconnected distributed systems with inherent and variable data transfer latencies. The data that the IEN receives from traffic control systems can be several seconds old. Data will age further by the time it reaches an external system through the IEN Web Server.

As described above, the timestamp that is included in IEN Outbound Web Service responses in the last_update element of the ienDeviceUpdate structure represents the last time that the IEN received an update for that device. The data in the responses reflect the status of the devices as of a few seconds earlier than the lastupdate timestamp.

4. INTERFACE OPERATIONS

The IEN Outbound Web Service provides the following operations:

- OP_Register
- OP_GetDeviceUpdate
- OP_UnRegister

These operations are defined in the ien.wsdl file (See Appendix A) and described in the following sections.

4.1 OP_REGISTER

Web clients use the OP_Register operation to register a connection session with the IEN Outbound Web Service. Web clients provide the name of the requesting Agency within their registration requests. The IEN Outbound Web Service responds back with a token, which web clients must use in their subsequent data requests, and a list of IEN organizations whose data is available to external systems. Only one connection session can be registered at a time under each Agency name. The web server will return an error if it receives a registration request with a name that is already registered (see Section 4.4 for more information).

Listing 4-1: Definition of OP_Register (from IEN.WSDL)

```

...
<wsdl:operation name="OP_Register">
  <wsdl:input message="tns:MSG_RegisterRequest"/>
  <wsdl:output message="tns:MSG_RegisterResponse"/>
  <wsdl:fault name="IENAlreadyConnectedFault" message="tns:IENAlreadyConnectedFault"/>
  <wsdl:fault name="IENTooManyConnectionsFault" message="tns:IENTooManyConnectionsFault"/>
  <wsdl:fault name="IENNotConfiguredFault" message="tns:IENNotConfiguredFault"/>
</wsdl:operation>
...
<wsdl:operation name="OP_Register">
  <soap:operation soapAction=""/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>

  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>

  <wsdl:fault name="IENAlreadyConnectedFault">
    <soap:fault name="IENAlreadyConnectedFault" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="IENTooManyConnectionsFault">
    <soap:fault name="IENTooManyConnectionsFault" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="IENNotConfiguredFault">
    <soap:fault name="IENNotConfiguredFault" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
...

```

4.2 INPUTS

The input message for the OP_Register operation is MSG_RegisterRequest, which contains the ien:registrationRequest element.

Listing 4-2: Definition of MSG_RegisterRequest (from IEN.WSDL)

```

...
<wsdl:message name="MSG_RegisterRequest">
  <wsdl:part name="request" element="ien:registrationRequest"/>
</wsdl:message>
...

```

RegistrationRequest contains a single element named “requestor”, which signifies the name of the requesting agency (example: “RIITS”). The IEN Outbound Web Service uses the requestor value to keep track of connection sessions. Only one connection session can be registered at a time with a particular requestor value.

Listing 4-3: Definition of RegistrationRequest (from IEN.XSD)

```

...
<xsd:element name="registrationRequest" type="tns:RegistrationRequest"/>
<xsd:complexType name="RegistrationRequest">
  <xsd:sequence>
    <xsd:element name="requestor" type="xsd:string" nillable="false"/>
  </xsd:sequence>
</xsd:complexType>
...

```

4.3 OUTPUTS

The output message for the OP_Register operation is MSG_RegisterResponse, which contains the ien:registrationResponse element.

Listing 4-4: Definition of MSG_RegisterResponse (from IEN.WSDL)

```

...
<wsdl:message name="MSG_RegisterResponse">
  <wsdl:part name="response" element="ien:registrationResponse"/>
</wsdl:message>
...

```

RegistrationResponse contains an IENOutboundResponse structure and token and organization elements.

Listing 4-5: Definition of RegistrationResponse (from IEN.XSD)

```

...
<xsd:element name="registrationResponse" type="tns:RegistrationResponse"/>
<xsd:complexType name="RegistrationResponse">

```

```

<xsd:complexContent>
  <xsd:extension base="tns:IENOutboundResponse">
    <xsd:sequence>
      <xsd:element name="token" type="xsd:string" nillable="false"/>
      <xsd:element name="organizations" type="tns:IENOrganizationInformation" minOccurs="0" maxOccurs="unbounded"
nillable="true"/>
    </xsd:sequence>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
...
<xsd:element name="ienOutboundResponse" type="tns:IENOutboundResponse"/>
<xsd:complexType name="IENOutboundResponse">
  <xsd:sequence>
    <xsd:element name="error" type="xsd:string" nillable="true"/>
    <xsd:element name="warning" type="xsd:string" nillable="true"/>
  </xsd:sequence>
</xsd:complexType>
...
<xsd:element name="ienOrganizationInformation" type="tns:IENOrganizationInformation"/>
<xsd:complexType name="IENOrganizationInformation">
  <xsd:sequence>
    <xsd:element name="organization_name" type="xsd:string" nillable="true"/>
    <xsd:element name="organization_function" type="xsd:string" nillable="true"/>
    <xsd:element name="organization_location" type="xsd:string" nillable="true"/>
    <xsd:element name="organization_id" type="xsd:string" nillable="false"/>
    <xsd:element name="organization_description" type="xsd:string" nillable="false"/>
  </xsd:sequence>
</xsd:complexType>
...

```

The IENOutboundResponse structure is used to report errors and warnings back to the web client. For example, the web service will report an error if it cannot read organization data from the IEN database.

The token element is a string that uniquely identifies a web client's connection session to the web service. This token must be included in OP_GetDataUpdateRequest (see Section 4.6) and OP_UnRegister (see Section 4.10) operations.

The organizations element contains information about each organization whose data is currently available through the web service. Agencies with multiple traffic control systems (such as Pasadena) will have multiple organization definitions. This information includes:

- organization_name: The organization name (example: "Pasadena DPW").
- organization_function: A free form description of the function of the organization (example: "Traffic Management Center").
- organization_location: A free form description of the location of the organization (example: "Pasadena TMC").
- organization_id: The identification string for the organization (examples: "2:1" and "5:1"). The string has the format of [number]:[number]1. Web clients use this value in OP_GetDataUpdateRequest (see Section 4.6) operations.

¹ Internally within the IEN, the first number represents the ID number of a deployment site. The second number represents the ID number of a traffic control system at that site (example: "2:1" represents the Pasadena TransCore Series 2000 traffic control system).

- **organization_description:** A free form description of the organization (examples: “Arcadia *TransSuite*® TCS signals” or “LACO KITS signals in San Gabriel Valley”).

Table 4-1 below lists the organizations that are expected to be available to external systems at the time of the initial deployment of the IEN Outbound Web Service:

Table 4-1: Available Organizations

ID	NAME	DESCRIPTION
2:1	Pasadena	Pasadena Series 2000 Signals
3:1	Pasadena	Pasadena QuicNet Signals
3:1	Pasadena	Pasadena i2TMS Signals
5:1	Arcadia	Arcadia <i>TransSuite</i> ® TCS Signals
6:1	Inglewood	Inglewood QuicNet Signals
7:1	West Hollywood	West Hollywood QuicNet Signals
8:1	Burbank	Burbank QuicNet Signals
9:1	Glendale	Glendale QuicNet Signals
10:1	Diamond Bar	Diamond Bar <i>TransSuite</i> ® TCS Signals
13:1	Gardena	Gardena QuicNet Signals
14:1	Downey	Downey i2TMS Signals

4.4 FAULTS

The OP_Register operation has the following faults:

Table 4-2: OP_Register Faults

FAULT	MEANING	CORRECTIVES
IENAlreadyConnectedFault	A registered connection already exists for the requestor given in MSG_RegisterRequest.	<ol style="list-style-type: none"> 1. If the token is known for the already active registration, use OP_GetDeviceUpdate to request data. 2. If not, allow the active registration to timeout. Registrations will time out after five minutes if no OP_GetDeviceUpdate operations are performed. Please note that the timeout period is configurable on the server and may be changed by an IEN Administrator.
IENTooManyConnectionsFault	The web service has reached its maximum number of active connections.	Contact LA County DPW.
IENNotConfiguredFault	The web service is mis-configured and cannot handle the registration request.	Contact LA County DPW.

4.5 OP_GETDEVICEUPDATE

Web clients use the OP_GetDeviceUpdate operation to request sets of data from the IEN Outbound Web Service. They do this by providing their connection token and an update specification in their request. The update specification identifies which types of data they want and from which organizations.

Listing 4-6: Definition of OP_GetDeviceUpdate (from IEN.WSDL)

```

...
<wsdl:operation name="OP_GetDeviceUpdate">
  <wsdl:input message="tns:MSG_GetDeviceUpdateRequest"/>
  <wsdl:output message="tns:MSG_GetDeviceUpdateResponse"/>
  <wsdl:fault name="IENUnknownConnectionFault" message="tns:IENUnknownConnectionFault"/>
</wsdl:operation>
...
<wsdl:operation name="OP_GetDeviceUpdate">
  <soap:operation soapAction=""/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>

  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>

  <wsdl:fault name="IENUnknownConnectionFault">
    <soap:fault name="IENUnknownConnectionFault" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
...

```

4.6 INPUTS

The input message for the OP_GetDeviceUpdate operation is MSG_GetDeviceUpdateRequest, which contains the ien:deviceUpdateRequest element.

Listing 4-7: Definition of MSG_GetDeviceUpdateRequest (from IEN.WSDL)

```

...
<wsdl:message name="MSG_GetDeviceUpdateRequest">
  <wsdl:part name="inventory_request" element="ien:deviceUpdateRequest"/>
</wsdl:message>
...

```

DeviceUpdateRequest contains an IENOutboundRequest structure and one or more UpdateSpec structures. The IENOutboundRequest structure contains the token, which is the connection identifier that was returned to the web client in the MSG_RegisterResponse message of the OP_Register operation (see Section 4.3).

Listing 4-8: Definition of DeviceUpdateRequest (from IEN.XSD)

```

...
<xsd:element name="deviceUpdateRequest" type="tns:DeviceUpdateRequest"/>
<xsd:complexType name="DeviceUpdateRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:IENOutboundRequest">
      <xsd:sequence>
        <xsd:element name="specs" type="tns:UpdateSpec" minOccurs="0" maxOccurs="unbounded" nillable="true"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
...
<xsd:element name="ienOutboundRequest" type="tns:IENOutboundRequest"/>
<xsd:complexType name="IENOutboundRequest">
  <xsd:sequence>
    <xsd:element name="token" type="xsd:string" nillable="false"/>
  </xsd:sequence>
</xsd:complexType>
...
<xsd:element name="updateSpec" type="tns:UpdateSpec"/>
<xsd:complexType name="UpdateSpec">
  <xsd:sequence>
    <xsd:element name="organization_id" type="xsd:string" nillable="true"/>
    <xsd:element name="update_types" type="tns:IENSupportedDeviceUpdate" minOccurs="1" maxOccurs="5" nillable="false"/>
  </xsd:sequence>
</xsd:complexType>
...

```

UpdateSpec structures contain the following elements:

- **organization_id:** The identification string of an organization to get data from. Corresponds to the organization_id value from the MSG_RegisterResponse message of the OP_Register operation (see Section 4.3). If this element is left empty, data for all available organization will be returned.
- **update_types:** The types of data to get from the specified organizations. One or more values from the IENSupportedDeviceUpdate enumeration. If this element is left empty, all supported update types will be returned.

Table 4-3 below lists the supported update types and the related data structures that are returned to the web client within the response:

Table 4-3: Update Types

UPDATE TYPE	RETURNED SCHEMA STRUCTURES
INTERSECTION_SIGNAL_CONFIG	IENSignalInventory
INTERSECTION_SIGNAL_SUMMARY	IENSignalSummary
INTERSECTION_SIGNAL_PHASES	IENPlannedPhases, IENLastCyclePhases
ARTERIAL_DETECTOR_CONFIG	IENDetectorInventory
ARTERIAL_DETECTOR_SUMMARY	IENDetectorSummary

These schema structures are described in Section 4.7.

4.7 OUTPUTS

The output message for the OP_GetDeviceUpdate operation is MSG_GetDeviceUpdateResponse which contains the ien:deviceUpdateResponse element.

Listing 4-9: Definition of MSG_GetDeviceUpdateResponse (from IEN.WSDL)

```

...
<wsdl:message name="MSG_GetDeviceUpdateResponse">
  <wsdl:part name="inventory" element="ien:deviceUpdateResponse"/>
</wsdl:message>
...

```

DeviceUpdateResponse contains an IENOutboundResponse structure and a set of device update structures that fulfill the specifications that were given in the DeviceUpdateRequest structure of the MSG_GetDeviceUpdateRequest message (see Section 4.6).

Listing 4-10: Definition of DeviceUpdateResponse (from IEN.XSD)

```

...
<xsd:element name="deviceUpdateResponse" type="tns:DeviceUpdateResponse"/>
<xsd:complexType name="DeviceUpdateResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:IENOutboundResponse">
      <xsd:sequence>
        <xsd:element name="signalInventory" type="tns:IENSignalInventory" minOccurs="0" maxOccurs="unbounded" nillable="true" />
        <xsd:element name="detectorInventory" type="tns:IENDetectorInventory" minOccurs="0" maxOccurs="unbounded" nillable="true" />
        <xsd:element name="signalSummary" type="tns:IENSignalSummary" minOccurs="0" maxOccurs="unbounded" nillable="true" />
        <xsd:element name="detectorSummary" type="tns:IENDetectorSummary" minOccurs="0" maxOccurs="unbounded" nillable="true" />
        <xsd:element name="lastCyclePhases" type="tns:IENLastCyclePhases" minOccurs="0" maxOccurs="unbounded" nillable="true" />
        <xsd:element name="plannedPhases" type="tns:IENPlannedPhases" minOccurs="0" maxOccurs="unbounded" nillable="true" />
        <xsd:element name="organization-information" type="tns:IENOrganizationInformation" minOccurs="0" maxOccurs="unbounded" nillable="true" />
        <xsd:element name="reporting-organizations" type="xsd:string" minOccurs="0" maxOccurs="unbounded" nillable="true" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
...

```

The IENOutboundResponse structure reports errors and warnings back to the web client. Some examples of these errors and warnings would be:

- Unable to query the database: The IEN Outbound Web Service could not read from the IEN database. Contact LA County.
- No updates for the specified organization: The specified organization is currently not reporting data to the IEN.

- Unsupported update type: `MSG_GetDeviceUpdateRequest` contained an invalid update specification.

The following structures contain the requested IEN intersection controller and arterial system detector data:

- `IENSignalInventory`
- `IENSignalSummary`
- `IENPlannedPhases`,
- `IENLastCyclePhases`
- `IENDetectorInventory`
- `IENDetectorSummary`

Each of the update type structures extend `IENDeviceUpdate`, which contains the following elements:

Table 4-4: IENDeviceUpdate Elements

NAME	MEANING
<code>organization_id</code>	The identification string of the owning organization. Corresponds to an <code>organization_id</code> value returned in the <code>IENOrganizationInformation</code> structure.
<code>device_id</code>	The identification number of the device within its native traffic control system.
<code>last_updated</code>	Time at which data for the device was last received by the IEN.

These update type structures are described in the following sections.

The `IENOrganizationInformation` and `reporting-organizations` structures are always provided in the `DeviceUpdateResponse`. `IENOrganizationInformation` (described in Section 4.3) contains information about Agencies whose data is available through the web service. This lets web clients know when new Agencies are made available. Not all of the Agencies that are available through the web service are reporting data to the IEN at any given time. Agencies that are currently reporting data to the IEN are listed in `reporting_organization` elements, whose values correspond to the `organization_id` of the `IENOrganizationInformation` structure.

4.7.1 IENSignalInventory

The `IENSignalInventory` structure contains the following elements:

Table 4-5: IENSignalInventory Elements

NAME	MEANING
<code>description</code>	Description of the intersection. Free-formatted string.
<code>signal_type</code>	Combination of hardware and firmware type. Free-formatted string.
<code>latitude</code>	Latitude of the intersection, in microdegrees (NAD83).
<code>longitude</code>	Longitude of the intersection, in microdegrees (NAD83).
<code>mainStreet</code>	Name of main street for the intersection. Free-formatted string.
<code>crossStreet</code>	Name of cross street for the intersection. Free-formatted string.

4.7.2 IENDetectorInventory

The IENDetectorInventory structure contains the following elements:

Table 4-6: IENDetectorInventory Elements

NAME	MEANING
associated_intersection_id	Identification number of the associated intersection.
averaging_period	Number of seconds over which averaged data was averaged.
roadway_name	Name of the roadway that contains the detector. Free-formatted string.
cross_street	Name of street that crosses the detector's roadway at its associated intersection. Free-formatted string.
direction	Direction of the traffic movement a particular detector is monitoring. A value from the IENDirection enumeration.
description	Description of the detector. Free-formatted string.

4.7.3 IENSignalSummary

The IENSignalSummary structure contains the following elements:

Table 4-7: IENSignalSummary Elements

NAME	MEANING
comm_state	Communications state of the intersection. A value from the IENSignalCommState enumeration.
timing_plan	ID number of the active traffic signal timing plan.
desired_cycle_length	Cycle length in seconds programmed for the timing plan that the controller is actually running.
desired_offset	Offset in seconds for the timing plan that the controller is currently running.
actual_offset	Offset in seconds that the controller ran for the last cycle.
signal_control_mode	Traffic control mode of the intersection. A value from the IENSignalControlMode enumeration.
signal_state	Operational state of the intersection. A value from the IENSignalState enumeration.

4.7.4 IENDetectorSummary

The IENDetectorSummary structure contains the following elements:

Table 4-8: IENDetectorSummary Elements

NAME	MEANING
state	Operational status of the device. A value from the IENDetectorState enumeration.
volume	Volume reported by the detector in vehicles per hour
occupancy	Percentage of time that the detector was occupied during the last reporting period (% expressed as an integer).
speed	Speed in miles per hour.

NAME	MEANING
avg_volume	Average volume (vehicles per hour) over the averaging period.
avg_occupancy	Average occupancy over the averaging period (% expressed as an integer).
avg_speed	Average speed (miles per hour) over the averaging period.

4.7.5 IENLastCyclePhases

The IENLastCyclePhases structure contains the following elements:

Table 4-9: IENLastCyclePhases Elements

NAME	MEANING
lastCycleLength	Number of seconds in the last cycle.
IENPhaseTimeDescription	A sequence of paired values that represent the identification number of a phase (phase_id) and the number of seconds that the phase was green during the last cycle (phase_time).

4.7.6 IENPlannedPhases

The IENPlannedPhases structure contains the following elements:

Table 4-10: IENPlannedPhases Elements

NAME	MEANING
IENPhaseTimeDescription	A sequence of paired values that represent the identification number of a phase (phase_id) and the maximum number of seconds for the phase to be green in the active timing plan (phase_time).

4.8 FAULTS

The OP_GetDeviceUpdate operation has the following faults:

Table 4-11: OP_GetDeviceUpdate Faults

FAULT	MEANING	CORRECTIVES
IENUnknownConnectionFault	No active connections match the given requestor and token values. The connection may have timed out or the web server may have been restarted.	Use the OP_Register operation to register with the web service.

4.9 OP_UNREGISTER

Web clients use the OP_UnRegister operation to unregister and end a connection session with the IEN Outbound Web Service. This signifies that they will not be requesting additional data. Web clients provide the name of the requesting Agency and current connection token within their unregistration requests. The web service responds back with any faults and closes the connection. The web service will automatically close a connection if no OP_GetDeviceUpdate requests are received for five minutes. Web clients will need to register with the web service again in order to request additional data. The request timeout period is configurable. External

Agencies should inform LA County if they will be requesting data at intervals greater than five minutes.

Listing 4-11: Definition of OP_UnRegister (from IEN.WSDL)

```

...
<wsdl:operation name="OP_UnRegister">
  <wsdl:input message="tns:MSG_UnRegisterRequest"/>
  <wsdl:output message="tns:MSG_UnRegisterResponse"/>
  <wsdl:fault name="IENUnknownConnectionFault" message="tns:IENUnknownConnectionFault"/>
</wsdl:operation>
...
<wsdl:operation name="OP_UnRegister">
  <soap:operation soapAction=""/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>

  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>

  <wsdl:fault name="IENUnknownConnectionFault">
    <soap:fault name="IENUnknownConnectionFault" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
...

```

4.10 INPUTS

The input message for the OP_UnRegister operation is MSG_UnRegisterRequest, which contains the ien:unregistrationRequest element.

Listing 4-12: Definition of MSG_UnRegisterRequest (from IEN.WSDL)

```

...
<wsdl:message name="MSG_UnRegisterRequest">
  <wsdl:part name="request" element="ien:unregistrationRequest"/>
</wsdl:message>
...

```

UnRegistrationRequest contains requestor and token elements. The requestor element is the name of the requesting Agency. The token, which is located within the IENOutboundRequest structure, is the connection identifier that was returned to the web client in the MSG_RegisterResponse message of the OP_Register operation (see Section 4.3).

Listing 4-13: Definition of UnRegistrationRequest (from IEN.XSD)

```

...
<xsd:element name="unregistrationRequest" type="tns:UnRegistrationRequest"/>
<xsd:complexType name="UnRegistrationRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:IENOutboundRequest">
      <xsd:sequence>
        <xsd:element name="requestor" type="xsd:string" nillable="false"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
...
<xsd:element name="ienOutboundRequest" type="tns:IENOutboundRequest"/>
<xsd:complexType name="IENOutboundRequest">
  <xsd:sequence>
    <xsd:element name="token" type="xsd:string" nillable="false"/>
  </xsd:sequence>
</xsd:complexType>
...

```

4.11 OUTPUTS

The output message for the OP_UnRegister operation is MSG_UnRegisterResponse, which contains the ien:unregistrationResponse element.

Listing 4-14: Definition of MSG_UnRegisterResponse (from IEN.WSDL)

```

...
<wsdl:message name="MSG_UnRegisterResponse">
  <wsdl:part name="response" element="ien:unregistrationResponse"/>
</wsdl:message>
...

```

UnRegistrationResponse contains an IENOutboundResponse structure to report errors and warnings back to the web client.

Listing 4-15: Definition of UnRegistrationResponse (from IEN.XSD)

```

...
<xsd:element name="unregistrationResponse" type="tns:UnRegistrationResponse"/>
<xsd:complexType name="UnRegistrationResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:IENOutboundResponse">
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
...
<xsd:element name="ienOutboundResponse" type="tns:IENOutboundResponse"/>
<xsd:complexType name="IENOutboundResponse">
  <xsd:sequence>
    <xsd:element name="error" type="xsd:string" nillable="true"/>
    <xsd:element name="warning" type="xsd:string" nillable="true"/>
  </xsd:sequence>
</xsd:complexType>
...

```

4.12 FAULTS

The OP_UnRegister operation has the following faults:

Table 4-12: OP_UnRegister Faults

FAULT	MEANING	CORRECTIVES
IENUnknownConnectionFault	No active connections match the given requestor and token values.	None required.

5. APPENDICES

5.1 APPENDIX A: IEN OUTBOUND WEB SERVICE WSDL

The IEN Outbound Web Service is defined by the WSDL file ien.wsdl, which is listed below. This listing is current as of the date of this document. The WSDL will be published on the IEN Web Server after implementation is complete.

Listing 5-1: IEN.WSDL

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

<wsdl:definitions name="IENOutboundWebService"
targetNamespace="http://ien.ladpw.org/OutboundInterface"
  xmlns:tns="http://ien.ladpw.org/OutboundInterface"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:ien="http://ien.ladpw.org/IEIEN">

  <wsdl:types>
    <xsd:schema elementFormDefault="qualified"
targetNamespace="http://ien.ladpw.org/OutboundInterface">
      <xsd:import namespace="http://ien.ladpw.org/IEIEN" schemaLocation="IEN.xsd"/>
    </xsd:schema>
  </wsdl:types>

  <wsdl:message name="MSG_RegisterRequest">
    <wsdl:part name="request" element="ien:registrationRequest"/>
  </wsdl:message>

  <wsdl:message name="MSG_RegisterResponse">
    <wsdl:part name="response" element="ien:registrationResponse"/>
  </wsdl:message>

  <wsdl:message name="MSG_UnRegisterRequest">
    <wsdl:part name="request" element="ien:unregistrationRequest"/>
  </wsdl:message>

  <wsdl:message name="MSG_UnRegisterResponse">
    <wsdl:part name="response" element="ien:unregistrationResponse"/>
  </wsdl:message>

  <wsdl:message name="MSG_GetDeviceUpdateRequest">
    <wsdl:part name="inventory_request" element="ien:deviceUpdateRequest"/>
  </wsdl:message>

  <wsdl:message name="MSG_GetDeviceUpdateResponse">
    <wsdl:part name="inventory" element="ien:deviceUpdateResponse"/>
  </wsdl:message>

  <wsdl:message name="IENAlreadyConnectedFault">
    <wsdl:part name="already_connected" element="ien:ienAlreadyConnected"/>
  </wsdl:message>
```

```

<wsdl:message name="IENTooManyConnectionsFault">
  <wsdl:part name="too_many_connections" element="ien:ienTooManyConnections"/>
</wsdl:message>

<wsdl:message name="IENNotConfiguredFault">
  <wsdl:part name="not_configured" element="ien:ienNotConfigured"/>
</wsdl:message>

<wsdl:message name="IENUnknownConnectionFault">
  <wsdl:part name="unknown_connection" element="ien:ienUnknownConnection"/>
</wsdl:message>

<wsdl:portType name="XMLOutboundInterface">
  <wsdl:operation name="OP_Register">
    <wsdl:input message="tns:MSG_RegisterRequest"/>
    <wsdl:output message="tns:MSG_RegisterResponse"/>
    <wsdl:fault name="IENAlreadyConnectedFault" message="tns:IENAlreadyConnectedFault"/>
    <wsdl:fault name="IENTooManyConnectionsFault"
message="tns:IENTooManyConnectionsFault"/>
    <wsdl:fault name="IENNotConfiguredFault" message="tns:IENNotConfiguredFault"/>
  </wsdl:operation>

  <wsdl:operation name="OP_UnRegister">
    <wsdl:input message="tns:MSG_UnRegisterRequest"/>
    <wsdl:output message="tns:MSG_UnRegisterResponse"/>
    <wsdl:fault name="IENUnknownConnectionFault" message="tns:IENUnknownConnectionFault"/>
  </wsdl:operation>

  <wsdl:operation name="OP_GetDeviceUpdate">
    <wsdl:input message="tns:MSG_GetDeviceUpdateRequest"/>
    <wsdl:output message="tns:MSG_GetDeviceUpdateResponse"/>
    <wsdl:fault name="IENUnknownConnectionFault" message="tns:IENUnknownConnectionFault"/>
  </wsdl:operation>
</wsdl:portType>

<wsdl:binding name="XMLOutboundInterfaceBinding" type="tns:XMLOutboundInterface">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

  <wsdl:operation name="OP_Register">
    <soap:operation soapAction=""/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>

    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>

    <wsdl:fault name="IENAlreadyConnectedFault">
      <soap:fault name="IENAlreadyConnectedFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="IENTooManyConnectionsFault">
      <soap:fault name="IENTooManyConnectionsFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="IENNotConfiguredFault">
      <soap:fault name="IENNotConfiguredFault" use="literal"/>
    </wsdl:fault>
  </wsdl:operation>

```

```
</wsdl:fault>
</wsdl:operation>

<wsdl:operation name="OP_UnRegister">
  <soap:operation soapAction=""/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>

  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>

  <wsdl:fault name="IENUnknownConnectionFault">
    <soap:fault name="IENUnknownConnectionFault" use="literal"/>
  </wsdl:fault>
</wsdl:operation>

<wsdl:operation name="OP_GetDeviceUpdate">
  <soap:operation soapAction=""/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>

  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>

  <wsdl:fault name="IENUnknownConnectionFault">
    <soap:fault name="IENUnknownConnectionFault" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
</wsdl:binding>

<wsdl:service name="IENOutboundInterface">
  <wsdl:port name="IENOutboundInterface" binding="tns:XMLOutboundInterfaceBinding">
    <soap:address location="http://localhost:8080/axis2/services/IENOutboundInterface"/>
  </wsdl:port>
</wsdl:service>

</wsdl:definitions>
```

5.2 APPENDIX B: IEN OUTBOUND WEB SERVICE XML SCHEMA

IEN Outbound Web Service message structures are described by the XML Schema file ien.xsd, which is listed below. This listing is current as of the date of this document. The schema will be published on the IEN Web Server after implementation is complete.

Listing 5-2: IEN.XSD

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

<xsd:schema targetNamespace="http://ien.ladpw.org/IEEN"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://ien.ladpw.org/IEEN"
  xmlns:tmdd="http://www.ite.org/tmdd">

  <xsd:simpleType name="IENSupportedDeviceUpdate">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="INTERSECTION_SIGNAL_INVENTORY"/>
      <xsd:enumeration value="INTERSECTION_SIGNAL_SUMMARY"/>
      <xsd:enumeration value="INTERSECTION_SIGNAL_PHASES"/>
      <xsd:enumeration value="ARTERIAL_DETECTOR_INVENTORY"/>
      <xsd:enumeration value="ARTERIAL_DETECTOR_SUMMARY"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="IENDirection">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="EastBound"/>
      <xsd:enumeration value="WestBound"/>
      <xsd:enumeration value="SouthBound"/>
      <xsd:enumeration value="NorthBound"/>
      <xsd:enumeration value="SouthEastBound"/>
      <xsd:enumeration value="SouthWestBound"/>
      <xsd:enumeration value="NorthEastBound"/>
      <xsd:enumeration value="NorthWestBound"/>
      <xsd:enumeration value="InBound"/>
      <xsd:enumeration value="OutBound"/>
      <xsd:enumeration value="East_West"/>
      <xsd:enumeration value="North_South"/>
      <xsd:enumeration value="InBound_and_Outbound"/>
      <xsd:enumeration value="NE_SW"/>
      <xsd:enumeration value="NW_SE"/>
      <xsd:enumeration value="None"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="IENSignalControlMode">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="FREE"/>
      <xsd:enumeration value="FIXED_TIME"/>
      <xsd:enumeration value="TIME_BASE_COORDINATION"/>
      <xsd:enumeration value="ACTUATED"/>
      <xsd:enumeration value="SEMI_ACTUATED"/>
      <xsd:enumeration value="CRITICAL_INTERSECTION_CONTROL"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>
```

```

        <xsd:enumeration value="TRAFFIC_RESPONSIVE"/>
        <xsd:enumeration value="ADAPTIVE"/>
        <xsd:enumeration value="TRANSITION"/>
        <xsd:enumeration value="EXTERNAL"/>
        <xsd:enumeration value="ATCS"/>
        <xsd:enumeration value="OTHER_NO_ADDITIONAL"/>
        <xsd:enumeration value="OTHER_ADDITIONAL"/>
        <xsd:enumeration value="UNKNOWN"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="IENSignalCommState">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="GOOD"/>
        <xsd:enumeration value="BAD"/>
        <xsd:enumeration value="UNKNOWN"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="IENDetectorState">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="FAILED"/>
        <xsd:enumeration value="OPERATIONAL"/>
        <xsd:enumeration value="OFF"/>
        <xsd:enumeration value="OTHER_NO_ADDITIONAL"/>
        <xsd:enumeration value="OTHER_ADDITIONAL"/>
        <xsd:enumeration value="UNKNOWN"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="IENSignalState">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="NORMAL_OPERATION"/>
        <xsd:enumeration value="FLASH"/>
        <xsd:enumeration value="PREEMPTION"/>
        <xsd:enumeration value="CONFLICT_FLASH"/>
        <xsd:enumeration value="FAILED"/>
        <xsd:enumeration value="OTHER_NO_ADDITIONAL"/>
        <xsd:enumeration value="OTHER_ADDITIONAL"/>
        <xsd:enumeration value="UNKNOWN"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:element name="ienOrganizationInformation" type="tns:IEOrganizationInformation"/>
<xsd:complexType name="IEOrganizationInformation">
    <xsd:sequence>
        <xsd:element name="organization_name" type="xsd:string" nillable="true"/>
        <xsd:element name="organization_function" type="xsd:string" nillable="true"/>
        <xsd:element name="organization_location" type="xsd:string" nillable="true"/>
        <xsd:element name="organization_id" type="xsd:string" nillable="false"/>
        <xsd:element name="organization_description" type="xsd:string" nillable="false"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="ienOutboundRequest" type="tns:IEOutboundRequest"/>
<xsd:complexType name="IEOutboundRequest">
    <xsd:sequence>

```

```

    <xsd:element name="token" type="xsd:string" nillable="false"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="ienOutboundResponse" type="tns:IENOutboundResponse"/>
<xsd:complexType name="IENOutboundResponse">
  <xsd:sequence>
    <xsd:element name="error" type="xsd:string" nillable="true"/>
    <xsd:element name="warning" type="xsd:string" nillable="true"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="registrationRequest" type="tns:RegistrationRequest"/>
<xsd:complexType name="RegistrationRequest">
  <xsd:sequence>
    <xsd:element name="requestor" type="xsd:string" nillable="false"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="registrationResponse" type="tns:RegistrationResponse"/>
<xsd:complexType name="RegistrationResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:IENOutboundResponse">
      <xsd:sequence>
        <xsd:element name="token" type="xsd:string" nillable="false"/>
        <xsd:element name="organizations" type="tns:IENOrganizationInformation" minOccurs="0"
maxOccurs="unbounded" nillable="true"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="unregistrationRequest" type="tns:UnRegistrationRequest"/>
<xsd:complexType name="UnRegistrationRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:IENOutboundRequest">
      <xsd:sequence>
        <xsd:element name="requestor" type="xsd:string" nillable="false"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="unregistrationResponse" type="tns:UnRegistrationResponse"/>
<xsd:complexType name="UnRegistrationResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:IENOutboundResponse">
      </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="updateSpec" type="tns:UpdateSpec"/>
<xsd:complexType name="UpdateSpec">
  <xsd:sequence>
    <xsd:element name="organization_id" type="xsd:string" nillable="true"/>
  </xsd:sequence>
</xsd:complexType>

```

```

        <xsd:element name="update_types" type="tns:IENSupportedDeviceUpdate"
minOccurs="1" maxOccurs="5" nillable="true"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:element name="deviceUpdateRequest" type="tns:DeviceUpdateRequest"/>
    <xsd:complexType name="DeviceUpdateRequest">
        <xsd:complexContent>
            <xsd:extension base="tns:IENOutboundRequest">
                <xsd:sequence>
                    <xsd:element name="specs" type="tns:UpdateSpec" minOccurs="0"
maxOccurs="unbounded" nillable="true"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:element name="ienDeviceUpdate" type="tns:IENDeviceUpdate"/>
    <xsd:complexType name="IENDeviceUpdate">
        <xsd:sequence>
            <xsd:element name="organization_id" type="xsd:string" nillable="false"/>
            <xsd:element name="device_id" type="xsd:int" nillable="false"/>
            <xsd:element name="last_update" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:element name="ienSignalInventory" type="tns:IENSignalInventory"/>
    <xsd:complexType name="IENSignalInventory">
        <xsd:complexContent>
            <xsd:extension base="tns:IENDeviceUpdate">
                <xsd:sequence>
                    <xsd:element name="description" type="xsd:string" nillable="true"/>
                    <xsd:element name="signal_type" type="xsd:string" nillable="true"/>
                    <xsd:element name="latitude" type="xsd:int" nillable="true"/>
                    <xsd:element name="longitude" type="xsd:int" nillable="true"/>
                    <xsd:element name="mainStreet" type="xsd:string" nillable="true"/>
                    <xsd:element name="crossStreet" type="xsd:string" nillable="true"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:element name="ienSignalSummary" type="tns:IENSignalSummary"/>
    <xsd:complexType name="IENSignalSummary">
        <xsd:complexContent>
            <xsd:extension base="tns:IENDeviceUpdate">
                <xsd:sequence>
                    <xsd:element name="comm_state" type="tns:IENSignalCommState"
nillable="true"/>
                    <xsd:element name="timing_plan" type="xsd:int" nillable="true"/>
                    <xsd:element name="desired_cycle_length" type="xsd:int"
nillable="true"/>
                    <xsd:element name="desired_offset" type="xsd:int" nillable="true"/>
                    <xsd:element name="actual_offset" type="xsd:int" nillable="true"/>
                    <xsd:element name="signal_control_mode"
type="tns:IENSignalControlMode" nillable="true"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

```

```

        <xsd:element name="signal_state" type="tns:IENSignalState"
nillable="true"/>
        </xsd:sequence>
    </xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:element name="ienDetectorInventory" type="tns:IENDetectorInventory"/>
<xsd:complexType name="IENDetectorInventory">
    <xsd:complexContent>
        <xsd:extension base="tns:IENDeviceUpdate">
            <xsd:sequence>
                <xsd:element name="associated_intersection_id" type="xsd:int"
nillable="true"/>
                <xsd:element name="averaging_period" type="xsd:int" nillable="true"/>
                <xsd:element name="roadway_name" type="xsd:string" nillable="true"/>
                <xsd:element name="cross_street" type="xsd:string" nillable="true"/>
                <xsd:element name="direction" type="tns:IENDirection" nillable="true"/>
                <xsd:element name="description" type="xsd:string" nillable="true"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:element name="ienDetectorSummary" type="tns:IENDetectorSummary"/>
<xsd:complexType name="IENDetectorSummary">
    <xsd:complexContent>
        <xsd:extension base="tns:IENDeviceUpdate">
            <xsd:sequence>
                <xsd:element name="state" type="tns:IENDetectorState" nillable="true"/>
                <xsd:element name="volume" type="xsd:int" nillable="true"/>
                <xsd:element name="occupancy" type="xsd:int" nillable="true"/>
                <xsd:element name="speed" type="xsd:int" nillable="true"/>
                <xsd:element name="avg_volume" type="xsd:int" nillable="true"/>
                <xsd:element name="avg_occupancy" type="xsd:int" nillable="true"/>
                <xsd:element name="avg_speed" type="xsd:int" nillable="true"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:element name="ienPhaseTimeDescription" type="tns:IENPhaseTimeDescription"/>
<xsd:complexType name="IENPhaseTimeDescription">
    <xsd:sequence>
        <xsd:element name="phase_id" type="xsd:short" nillable="false"/>
        <xsd:element name="phase_time" type="xsd:short" nillable="false"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="ienPlannedPhases" type="tns:IENPlannedPhases"/>
<xsd:complexType name="IENPlannedPhases">
    <xsd:complexContent>
        <xsd:extension base="tns:IENDeviceUpdate">
            <xsd:sequence>
                <xsd:element name="phases" type="tns:IENPhaseTimeDescription"
minOccurs="0" maxOccurs="unbounded" nillable="true"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

    <xsd:element name="ienLastCyclePhases" type="tns:IENLastCyclePhases"/>
<xsd:complexType name="IENLastCyclePhases">
    <xsd:complexContent>
        <xsd:extension base="tns:IENDeviceUpdate">
            <xsd:sequence>
                <xsd:element name="lastCycleLength" type="xsd:int" nillable="true"/>
                <xsd:element name="greens" type="tns:IENPhaseTimeDescription"
minOccurs="0" maxOccurs="unbounded" nillable="true"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:element name="deviceUpdateResponse" type="tns:DeviceUpdateResponse"/>
<xsd:complexType name="DeviceUpdateResponse">
    <xsd:complexContent>
        <xsd:extension base="tns:IENOutboundResponse">
            <xsd:sequence>
                <xsd:element name="signalInventory" type="tns:IENSignalInventory"
minOccurs="0" maxOccurs="unbounded" nillable="true" />
                <xsd:element name="detectorInventory" type="tns:IENDetectorInventory" minOccurs="0"
maxOccurs="unbounded" nillable="true" />
                <xsd:element name="signalSummary" type="tns:IENSignalSummary" minOccurs="0"
maxOccurs="unbounded" nillable="true"/>
                <xsd:element name="detectorSummary" type="tns:IENDetectorSummary"
minOccurs="0" maxOccurs="unbounded" nillable="true" />
                <xsd:element name="lastCyclePhases" type="tns:IENLastCyclePhases" minOccurs="0"
maxOccurs="unbounded" nillable="true" />
                <xsd:element name="plannedPhases" type="tns:IENPlannedPhases" minOccurs="0"
maxOccurs="unbounded" nillable="true"/>
                <xsd:element name="organization-information" type="tns:IENOrganizationInformation"
minOccurs="0" maxOccurs="unbounded" nillable="true"/>
                <xsd:element name="reporting-organizations" type="xsd:string" minOccurs="0"
maxOccurs="unbounded" nillable="true"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:element name="ienOutboundFault" type="tns:IENOutboundFault"/>
<xsd:complexType name="IENOutboundFault">
    <xsd:sequence>
        <xsd:element name="message" type="xsd:string" nillable="false"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="ienAlreadyConnected" type="tns:IENAlreadyConnected"/>
<xsd:complexType name="IENAlreadyConnected">
    <xsd:complexContent>
        <xsd:extension base="tns:IENOutboundFault">
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```



```
<xsd:element name="ienTooManyConnections" type="tns:IENTooManyConnections"/>
<xsd:complexType name="IENTooManyConnections">
  <xsd:complexContent>
    <xsd:extension base="tns:IENOutboundFault">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="ienNotConfigured" type="tns:IENNotConfigured"/>
<xsd:complexType name="IENNotConfigured">
  <xsd:complexContent>
    <xsd:extension base="tns:IENOutboundFault">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="ienUnknownConnection" type="tns:IENUnknownConnection"/>
<xsd:complexType name="IENUnknownConnection">
  <xsd:complexContent>
    <xsd:extension base="tns:IENOutboundFault">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

</xsd:schema>
```

5.3 APPENDIX C: EXAMPLE IEN OUTBOUND WEB SERVICE REQUESTS AND RESPONSES

Example IEN Outbound Web Service request and response messages are provided below.

5.3.1 OP_Register

5.3.1.1 Request

```
<ien:registrationRequest xmlns:ien="http://ien.ladpw.org/IEN"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <requestor>TestClient</requestor>
</ien:registrationRequest>
```

5.3.1.2 Response

```
<registrationResponse xmlns="http://ien.ladpw.org/IEN">
  <token xmlns="">2010-06-15-TestClient-13:36:36</token>
  <organizations xmlns="">
    <organization_name>Pasadena</organization_name>
    <organization_function>Traffic Managment Center</organization_function>
    <organization_location>Pasadena QN & I2</organization_location>
    <organization_id>3:2</organization_id>
    <organization_description>City of Pasadena TMC</organization_description>
  </organizations>
  <organizations xmlns="">
    <organization_name>Pasadena</organization_name>
    <organization_function>Traffic Managment Center</organization_function>
    <organization_location>Pasadena QN & I2</organization_location>
    <organization_id>3:1</organization_id>
    <organization_description>City of Pasadena TMC</organization_description>
  </organizations>
  <organizations xmlns="">
    <organization_name>Inglewood</organization_name>
    <organization_function>Traffic Managment Center</organization_function>
    <organization_location>Inglewood</organization_location>
    <organization_id>6:1</organization_id>
    <organization_description>City of Inglewood TMC</organization_description>
  </organizations>
  <organizations xmlns="">
    <organization_name>Burbank</organization_name>
    <organization_function>Traffic Managment Center</organization_function>
    <organization_location>Burbank</organization_location>
    <organization_id>8:1</organization_id>
    <organization_description>City of Burbank TMC</organization_description>
  </organizations>
  <organizations xmlns="">
    <organization_name>West Hollywood</organization_name>
    <organization_function>Traffic Managment Center</organization_function>
    <organization_location>West Hollywood</organization_location>
    <organization_id>7:1</organization_id>
    <organization_description>West Hollywood TMC</organization_description>
  </organizations>
  <organizations xmlns="">
    <organization_name>Gardena</organization_name>
```

```

    <organization_function>Traffic Managment Center</organization_function>
    <organization_location>Gardena</organization_location>
    <organization_id>13:1</organization_id>
    <organization_description>City of Gardena TMC</organization_description>
</organizations>
<organizations xmlns="">
    <organization_name>Pasadena</organization_name>
    <organization_function>Traffic Managment Center</organization_function>
    <organization_location>Pasadena Series 2000</organization_location>
    <organization_id>2:1</organization_id>
    <organization_description>City of Pasadena TMC</organization_description>
</organizations>
<organizations xmlns="">
    <organization_name>Glendale</organization_name>
    <organization_function>Traffic Managment Center</organization_function>
    <organization_location>Glendale</organization_location>
    <organization_id>9:1</organization_id>
    <organization_description>City of Glendale TMC</organization_description>
</organizations>
<organizations xmlns="">
    <organization_name>Arcadia</organization_name>
    <organization_function>Traffic Managment Center</organization_function>
    <organization_location>Arcadia</organization_location>
    <organization_id>5:1</organization_id>
    <organization_description>City of Arcadia TMC</organization_description>
</organizations>
<organizations xmlns="">
    <organization_name>Downey</organization_name>
    <organization_function>Traffic Managment Center</organization_function>
    <organization_location>Downey</organization_location>
    <organization_id>14:1</organization_id>
    <organization_description>City of Downey TMC</organization_description>
</organizations>
<organizations xmlns="">
    <organization_name>Diamond Bar</organization_name>
    <organization_function>Traffic Managment Center</organization_function>
    <organization_location>Diamond Bar</organization_location>
    <organization_id>10:1</organization_id>
    <organization_description>City of Diamond Bar TMC</organization_description>
</organizations>
</registrationResponse>

```

5.3.2 OP_GetDeviceUpdate

5.3.2.1 Request

```

<ien:deviceUpdateRequest                                xmlns:ien="http://ien.ladpw.org/IEN"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <token>2010-06-15-TestClient-13:36:36</token>
  <specs>
    <organization_id>3:2</organization_id>
    <update_types>ARTERIAL_DETECTOR_INVENTORY</update_types>
    <update_types>ARTERIAL_DETECTOR_SUMMARY</update_types>
    <update_types>INTERSECTION_SIGNAL_INVENTORY</update_types>

```

```
<update_types>INTERSECTION_SIGNAL_SUMMARY</update_types>
<update_types>INTERSECTION_SIGNAL_PHASES</update_types>
</specs>
<specs>
  <organization_id>3:1</organization_id>
  <update_types>ARTERIAL_DETECTOR_INVENTORY</update_types>
  <update_types>ARTERIAL_DETECTOR_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_INVENTORY</update_types>
  <update_types>INTERSECTION_SIGNAL_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_PHASES</update_types>
</specs>
<specs>
  <organization_id>6:1</organization_id>
  <update_types>ARTERIAL_DETECTOR_INVENTORY</update_types>
  <update_types>ARTERIAL_DETECTOR_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_INVENTORY</update_types>
  <update_types>INTERSECTION_SIGNAL_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_PHASES</update_types>
</specs>
<specs>
  <organization_id>8:1</organization_id>
  <update_types>ARTERIAL_DETECTOR_INVENTORY</update_types>
  <update_types>ARTERIAL_DETECTOR_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_INVENTORY</update_types>
  <update_types>INTERSECTION_SIGNAL_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_PHASES</update_types>
</specs>
<specs>
  <organization_id>7:1</organization_id>
  <update_types>ARTERIAL_DETECTOR_INVENTORY</update_types>
  <update_types>ARTERIAL_DETECTOR_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_INVENTORY</update_types>
  <update_types>INTERSECTION_SIGNAL_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_PHASES</update_types>
</specs>
<specs>
  <organization_id>13:1</organization_id>
  <update_types>ARTERIAL_DETECTOR_INVENTORY</update_types>
  <update_types>ARTERIAL_DETECTOR_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_INVENTORY</update_types>
  <update_types>INTERSECTION_SIGNAL_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_PHASES</update_types>
</specs>
<specs>
  <organization_id>2:1</organization_id>
  <update_types>ARTERIAL_DETECTOR_INVENTORY</update_types>
  <update_types>ARTERIAL_DETECTOR_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_INVENTORY</update_types>
  <update_types>INTERSECTION_SIGNAL_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_PHASES</update_types>
</specs>
<specs>
  <organization_id>9:1</organization_id>
```

```

    <update_types>ARTERIAL_DETECTOR_INVENTORY</update_types>
    <update_types>ARTERIAL_DETECTOR_SUMMARY</update_types>
    <update_types>INTERSECTION_SIGNAL_INVENTORY</update_types>
    <update_types>INTERSECTION_SIGNAL_SUMMARY</update_types>
    <update_types>INTERSECTION_SIGNAL_PHASES</update_types>
  </specs>
</specs>
  <organization_id>5:1</organization_id>
  <update_types>ARTERIAL_DETECTOR_INVENTORY</update_types>
  <update_types>ARTERIAL_DETECTOR_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_INVENTORY</update_types>
  <update_types>INTERSECTION_SIGNAL_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_PHASES</update_types>
</specs>
</specs>
  <organization_id>14:1</organization_id>
  <update_types>ARTERIAL_DETECTOR_INVENTORY</update_types>
  <update_types>ARTERIAL_DETECTOR_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_INVENTORY</update_types>
  <update_types>INTERSECTION_SIGNAL_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_PHASES</update_types>
</specs>
</specs>
  <organization_id>10:1</organization_id>
  <update_types>ARTERIAL_DETECTOR_INVENTORY</update_types>
  <update_types>ARTERIAL_DETECTOR_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_INVENTORY</update_types>
  <update_types>INTERSECTION_SIGNAL_SUMMARY</update_types>
  <update_types>INTERSECTION_SIGNAL_PHASES</update_types>
</specs>
</ien:deviceUpdateRequest>

```

5.3.2.2 Response

```

<deviceUpdateResponse xmlns="http://ien.ladpw.org/IEN">
  <error xmlns=""/>
  <warning xmlns="">Org Glendale has no updates now.Org Burbank has no updates
now.</warning>
  <signalInventory xmlns="">
    <organization_id>3:2</organization_id>
    <device_id>275</device_id>
    <last_update>06/15/2010 13:38:00</last_update>
    <description>Del Mar Blvd @ Orange Grove Blvd</description>
    <signal_type>Bi Tran 203?</signal_type>
    <latitude>34140731</latitude>
    <longitude>-118159456</longitude>
    <mainStreet/>
    <crossStreet/>
  </signalInventory>
  <signalInventory xmlns="">
    <organization_id>3:2</organization_id>
    <device_id>274</device_id>
    <last_update>06/15/2010 13:38:00</last_update>
    <description>Cordova St @ Hill Ave</description>

```

```

    <signal_type>Bi Tran 203?</signal_type>
    <latitude>34142654</latitude>
    <longitude>-118121308</longitude>
    <mainStreet/>
    <crossStreet/>
</signalInventory>
...
<detectorInventory xmlns="">
  <organization_id>7:1</organization_id>
  <device_id>6008</device_id>
  <last_update>06/15/2010 13:38:09</last_update>
  <averaging_period>300</averaging_period>
  <roadway_name>Unknown</roadway_name>
  <direction>None</direction>
  <description>None:Unknown:Lane 1:Type DT_INDUCTIVE_LOOP</description>
</detectorInventory>
<detectorInventory xmlns="">
  <organization_id>7:1</organization_id>
  <device_id>6009</device_id>
  <last_update>06/15/2010 13:38:09</last_update>
  <averaging_period>300</averaging_period>
  <roadway_name>Unknown</roadway_name>
  <direction>None</direction>
  <description>None:Unknown:Lane 1:Type DT_INDUCTIVE_LOOP</description>
</detectorInventory>
...
<signalSummary xmlns="">
  <organization_id>3:2</organization_id>
  <device_id>288</device_id>
  <last_update>06/15/2010 13:38:04</last_update>
  <comm_state>GOOD</comm_state>
  <timing_plan>3</timing_plan>
  <desired_cycle_length>70</desired_cycle_length>
  <desired_offset>6</desired_offset>
  <actual_offset>62</actual_offset>
  <signal_control_mode>TIME_BASE_COORDINATION</signal_control_mode>
  <signal_state>NORMAL_OPERATION</signal_state>
</signalSummary>
<signalSummary xmlns="">
  <organization_id>3:2</organization_id>
  <device_id>289</device_id>
  <last_update>06/15/2010 13:38:04</last_update>
  <comm_state>GOOD</comm_state>
  <timing_plan>3</timing_plan>
  <desired_cycle_length>70</desired_cycle_length>
  <desired_offset>11</desired_offset>
  <actual_offset>58</actual_offset>
  <signal_control_mode>TIME_BASE_COORDINATION</signal_control_mode>
  <signal_state>NORMAL_OPERATION</signal_state>
</signalSummary>
...
<detectorSummary xmlns="">
  <organization_id>10:1</organization_id>

```

```

<device_id>10121</device_id>
<last_update>06/15/2010 13:38:31</last_update>
<state>OPERATIONAL</state>
<volume>0</volume>
<occupancy>0</occupancy>
<speed>0</speed>
<avg_volume>130</avg_volume>
<avg_occupancy>1</avg_occupancy>
<avg_speed>16</avg_speed>
</detectorSummary>
<detectorSummary xmlns="">
  <organization_id>10:1</organization_id>
  <device_id>100406</device_id>
  <last_update>06/15/2010 13:38:29</last_update>
  <state>OPERATIONAL</state>
  <volume>180</volume>
  <occupancy>0</occupancy>
  <speed>0</speed>
  <avg_volume>179</avg_volume>
  <avg_occupancy>1</avg_occupancy>
  <avg_speed>1</avg_speed>
</detectorSummary>
...
<lastCyclePhases xmlns="">
  <organization_id>10:1</organization_id>
  <device_id>705</device_id>
  <last_update>06/15/2010 14:50:22</last_update>
  <lastCycleLength>110</lastCycleLength>
  <greens>
    <phase_id>1</phase_id>
    <phase_time>13</phase_time>
  </greens>
  <greens>
    <phase_id>2</phase_id>
    <phase_time>85</phase_time>
  </greens>
  <greens>
    <phase_id>3</phase_id>
    <phase_time>0</phase_time>
  </greens>
  <greens>
    <phase_id>4</phase_id>
    <phase_time>0</phase_time>
  </greens>
  <greens>
    <phase_id>5</phase_id>
    <phase_time>5</phase_time>
  </greens>
  <greens>
    <phase_id>6</phase_id>
    <phase_time>99</phase_time>
  </greens>
  <greens>

```

```

    <phase_id>7</phase_id>
    <phase_time>0</phase_time>
  </greens>
</greens>
  <phase_id>8</phase_id>
  <phase_time>0</phase_time>
</greens>
</lastCyclePhases>
<lastCyclePhases xmlns="">
  <organization_id>10:1</organization_id>
  <device_id>704</device_id>
  <last_update>06/15/2010 14:50:31</last_update>
  <lastCycleLength>110</lastCycleLength>
  <greens>
    <phase_id>1</phase_id>
    <phase_time>0</phase_time>
  </greens>
  <greens>
    <phase_id>2</phase_id>
    <phase_time>58</phase_time>
  </greens>
  <greens>
    <phase_id>3</phase_id>
    <phase_time>20</phase_time>
  </greens>
  <greens>
    <phase_id>4</phase_id>
    <phase_time>7</phase_time>
  </greens>
  <greens>
    <phase_id>5</phase_id>
    <phase_time>0</phase_time>
  </greens>
  <greens>
    <phase_id>6</phase_id>
    <phase_time>58</phase_time>
  </greens>
  <greens>
    <phase_id>7</phase_id>
    <phase_time>0</phase_time>
  </greens>
  <greens>
    <phase_id>8</phase_id>
    <phase_time>31</phase_time>
  </greens>
</lastCyclePhases>
...
<plannedPhases xmlns="">
  <organization_id>10:1</organization_id>
  <device_id>603</device_id>
  <last_update>06/15/2010 14:50:21</last_update>
  <phases>
    <phase_id>1</phase_id>

```



```

    <phase_time>14</phase_time>
  </phases>
</phases>
  <phase_id>2</phase_id>
  <phase_time>46</phase_time>
</phases>
</phases>
  <phase_id>3</phase_id>
  <phase_time>0</phase_time>
</phases>
</phases>
  <phase_id>4</phase_id>
  <phase_time>19</phase_time>
</phases>
</phases>
  <phase_id>5</phase_id>
  <phase_time>14</phase_time>
</phases>
</phases>
  <phase_id>6</phase_id>
  <phase_time>46</phase_time>
</phases>
</phases>
  <phase_id>7</phase_id>
  <phase_time>0</phase_time>
</phases>
</phases>
  <phase_id>8</phase_id>
  <phase_time>19</phase_time>
</phases>
</plannedPhases>
<plannedPhases xmlns="">
  <organization_id>10:1</organization_id>
  <device_id>602</device_id>
  <last_update>06/15/2010 14:50:21</last_update>
  <phases>
    <phase_id>1</phase_id>
    <phase_time>14</phase_time>
  </phases>
  <phases>
    <phase_id>2</phase_id>
    <phase_time>56</phase_time>
  </phases>
  <phases>
    <phase_id>3</phase_id>
    <phase_time>0</phase_time>
  </phases>
  <phases>
    <phase_id>4</phase_id>
    <phase_time>14</phase_time>
  </phases>
  <phases>
    <phase_id>5</phase_id>

```

```

    <phase_time>14</phase_time>
  </phases>
</phases>
  <phase_id>6</phase_id>
  <phase_time>56</phase_time>
</phases>
</phases>
  <phase_id>7</phase_id>
  <phase_time>0</phase_time>
</phases>
</phases>
  <phase_id>8</phase_id>
  <phase_time>14</phase_time>
</phases>
</plannedPhases>
...
<organization-information xmlns="">
  <organization_name>Pasadena</organization_name>
  <organization_function>Traffic Management Center</organization_function>
  <organization_location>Pasadena QN & I2</organization_location>
  <organization_id>3:2</organization_id>
  <organization_description>City of Pasadena TMC</organization_description>
</organization-information>
<organization-information xmlns="">
  <organization_name>Pasadena</organization_name>
  <organization_function>Traffic Management Center</organization_function>
  <organization_location>Pasadena QN & I2</organization_location>
  <organization_id>3:1</organization_id>
  <organization_description>City of Pasadena TMC</organization_description>
</organization-information>
<organization-information xmlns="">
  <organization_name>Inglewood</organization_name>
  <organization_function>Traffic Management Center</organization_function>
  <organization_location>Inglewood</organization_location>
  <organization_id>6:1</organization_id>
  <organization_description>City of Inglewood TMC</organization_description>
</organization-information>
<organization-information xmlns="">
  <organization_name>Burbank</organization_name>
  <organization_function>Traffic Management Center</organization_function>
  <organization_location>Burbank</organization_location>
  <organization_id>8:1</organization_id>
  <organization_description>City of Burbank TMC</organization_description>
</organization-information>
<organization-information xmlns="">
  <organization_name>West Hollywood</organization_name>
  <organization_function>Traffic Management Center</organization_function>
  <organization_location>West Hollywood</organization_location>
  <organization_id>7:1</organization_id>
  <organization_description>West Hollywood TMC</organization_description>
</organization-information>
<organization-information xmlns="">
  <organization_name>Gardena</organization_name>

```

```

    <organization_function>Traffic Managment Center</organization_function>
    <organization_location>Gardena</organization_location>
    <organization_id>13:1</organization_id>
    <organization_description>City of Gardena TMC</organization_description>
</organization-information>
<organization-information xmlns="">
    <organization_name>Pasadena</organization_name>
    <organization_function>Traffic Managment Center</organization_function>
    <organization_location>Pasadena Series 2000</organization_location>
    <organization_id>2:1</organization_id>
    <organization_description>City of Pasadena TMC</organization_description>
</organization-information>
<organization-information xmlns="">
    <organization_name>Glendale</organization_name>
    <organization_function>Traffic Managment Center</organization_function>
    <organization_location>Glendale</organization_location>
    <organization_id>9:1</organization_id>
    <organization_description>City of Glendale TMC</organization_description>
</organization-information>
<organization-information xmlns="">
    <organization_name>Arcadia</organization_name>
    <organization_function>Traffic Managment Center</organization_function>
    <organization_location>Arcadia</organization_location>
    <organization_id>5:1</organization_id>
    <organization_description>City of Arcadia TMC</organization_description>
</organization-information>
<organization-information xmlns="">
    <organization_name>Downey</organization_name>
    <organization_function>Traffic Managment Center</organization_function>
    <organization_location>Downey</organization_location>
    <organization_id>14:1</organization_id>
    <organization_description>City of Downey TMC</organization_description>
</organization-information>
<organization-information xmlns="">
    <organization_name>Diamond Bar</organization_name>
    <organization_function>Traffic Managment Center</organization_function>
    <organization_location>Diamond Bar</organization_location>
    <organization_id>10:1</organization_id>
    <organization_description>City of Diamond Bar TMC</organization_description>
</organization-information>
<reporting-organizations xmlns="">6:1</reporting-organizations>
<reporting-organizations xmlns="">13:1</reporting-organizations>
<reporting-organizations xmlns="">5:1</reporting-organizations>
<reporting-organizations xmlns="">7:1</reporting-organizations>
<reporting-organizations xmlns="">2:1</reporting-organizations>
<reporting-organizations xmlns="">10:1</reporting-organizations>
<reporting-organizations xmlns="">14:1</reporting-organizations>
<reporting-organizations xmlns="">3:2</reporting-organizations>
</deviceUpdateResponse>

```

5.3.3 OP_UnRegister

5.3.3.1 Request

```
<ien:unregistrationRequest xmlns:ien="http://ien.ladpw.org/IEN"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <token>2010-06-15-TestClient-13:36:36</token>
  <requestor>TestClient</requestor>
</ien:unregistrationRequest>
```

5.3.3.2 Response

```
<unregistrationResponse xmlns="http://ien.ladpw.org/IEN"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <error xsi:nil="true" xmlns=""/>
  <warning xsi:nil="true" xmlns=""/>
</unregistrationResponse>
```

5.4 APPENDIX D: LOCATION OF REQUIRED DATA IN THE XML SCHEMA

Table 5-1 shows where required intersection controller data are located within the XML schema:

Table 5-1: Intersection Controller Data

IEN DATA		IEN XML SCHEMA	
ATTRIBUTE	DESCRIPTION	ELEMENT	
Intersection Id	Unique identifier for the device within its native traffic control system.	IENDeviceUpdate	device_id
Update time	Time at which data for the device was last received by the IEN.		last_update
Organization Id	The owning organization of the device.		organization_id
Intersection Description	Free form description of intersection.	IENSignalInventory	description
Intersection Controller Type	Combination of hardware and firmware type.		signal_type
Intersection Longitude	Longitude of the intersection, in microdegrees.		latitude
Intersection Latitude	Latitude of the intersection, in microdegrees.		longitude
Intersection Main Street	Main street of intersection.		mainStreet
Intersection Cross Street	Cross street of intersection.		crossStreet
Intersection control mode	Traffic control mode of the intersection.	IENSignalSummary	signal_control_mode
Intersection signal state	Operational state of the intersection.		signal_state
Timing plan Id	ID number of the active traffic signal timing plan.		timing_plan
Desired cycle length	Cycle length in seconds programmed for the timing plan that the controller is actually running.		desired_cycle_length
Desired offset	Offset in seconds for the timing plan that the controller is currently running.		desired_offset
Actual offset	Offset run by the intersection for the last cycle.		actual_offset
Comm. State	Communication state of the intersection.		comm_state
Planned phase times	Planned phase times for all possible phases for the controller.	IENPlannedPhases	phases
Last Cycle Cycle Length	Length (in seconds) of the last cycle.	IENLastCyclePhases	lastCycleLength

IEN DATA		IEN XML SCHEMA	
ATTRIBUTE	DESCRIPTION	ELEMENT	
Last cycle phase times	Length of all defined phases in the controller's previous cycle.		greens

Table 5-2 shows where required arterial system detector data are located within the XML schema:

Table 5-2: Arterial System Detector Data

IEN DATA		IEN XML SCHEMA	
ATTRIBUTE	DESCRIPTION	DATA ELEMENT	
Detector Id	Unique identifier for the device within its native traffic control system.	IENDeviceUpdate	device_id
Update time	Time at which data for the device was last received by the IEN.		last_update
Organization Id	The owning organization of the device.		organization_id
Detector Direction	Direction of the traffic movement a particular detector is monitoring.	IENDetectorInventory	direction
Detector Data Averaging period	Number of seconds over which averaged data was averaged.		average_period
Detector Roadway Name	Name of the roadway that contains the detector.		roadway_name
Detector Cross Street	Name of the street that crosses the detector's roadway at its associated intersection.		cross_street
Associated intersection	The intersection with which the detector is associated.		associated_intersection_id
Description	Free form description of the detector.		description
Detector State	Operational status of the device.	IENDetectorSummary	state
Detector Volume	Volume reported by the detector in vehicles per hour.		volume
Average Detector Volume	Average volume in vehicles per hour.		avg_volume
Detector Occupancy	Percentage of time that the detector was occupied during the last reporting period.		occupancy

IEN DATA		IEN XML SCHEMA	
ATTRIBUTE	DESCRIPTION	DATA ELEMENT	
Averaged Detector Occupancy	Average occupancy over the averaging period.		avg_occupancy
Detector Speed	Speed in miles per hour.		speed
Averaged Detector Speed	Average speed (miles per hour) over the averaging period.		avg_speed

5.5 APPENDIX E: TRAFFIC CONTROL SYSTEM SUPPORT FOR IEN DATA

Table 5-3 below shows which types of traffic control systems are used by IEN member Agencies.

Table 5-3: Organization Traffic Control Systems

ORG#	ORGANIZATION	TRAFFIC CONTROL SYSTEM
2:1	Pasadena	TransCore Series 2000
3:1	Pasadena	McCain QuicNet
3:2	Pasadena	Siemens I2TMS
5:1	Arcadia	TransCore <i>TransSuite</i> TCS
6:1	Inglewood	McCain QuicNet
7:1	West Hollywood	McCain QuicNet
8:1	Burbank	McCain QuicNet
9:1	Glendale	McCain QuicNet
10:1	Diamond Bar	TransCore <i>TransSuite</i> TCS
13:1	Gardena	McCain QuicNet
14:1	Downey	Siemens I2TMS
15:1	Santa Clarita	McCain QuicNet

Table 5-4 below shows which IEN intersection data are supported by each traffic control system.

Table 5-4: Traffic Control System Support for IEN Intersection Data

DATA	MCCAIN QUICNET	TRANSCORE SERIES 2000	TRANSCORE <i>TRANSSUITE</i>	SIEMENS I2TMS
description	Yes	Yes	Yes	Yes
signal_type	Yes	Yes	Yes	Yes
latitude	Yes	Yes	Yes	Yes
longitude	Yes	Yes	Yes	Yes
mainStreet	No	No	Yes	No
crossStreet	No	No	Yes	No
comm_state	Yes	Yes	Yes	Yes
timing_plan	Yes	Yes	Yes	Yes
desired_cycle_length	Yes	Yes	Yes	Yes
desired_offset	Yes	Yes	Yes	Yes
actual_offset	Yes ²	Yes	Yes	Yes
signal_control_mode	Yes	Yes	Yes	Yes
signal_state	Yes	Yes	Yes	Yes

² Requires Bi Tran 233RV2 firmware and adequate field communications.

DATA	MCCAIN QUICNET	TRANSCORE SERIES 2000	TRANSCORE TRANSSUITE	SIEMENS I2TMS
lastCycleLength	Yes	Yes	Yes	Yes
Last cycle phase green times	Yes	Yes	Yes	Yes
Planned phase max green times	Yes ³	Yes	Yes	Yes

5 below shows which IEN detector data are supported by each traffic control system.

Table 5-5: Traffic Control System Support for IEN Detector Data

DATA	MCCAIN QUICNET	TRANSCORE SERIES 2000	TRANSCORE TRANSSUITE	SIEMENS I2TMS
associated_intersection_id ⁴	No	No	No	No
averaging_period	Yes	Yes	Yes	Yes
roadway_name	No	No	No	Yes
cross_street ⁴	No	No	No	No
direction	Yes	No	Yes	Yes
description ⁴	No	No	No	No
state	Yes	Yes	Yes	Yes
volume	Yes	Yes	Yes	Yes
occupancy	Yes	Yes	Yes	Yes
speed	No	Yes	Yes	Yes
avg_volume	Yes	Yes	Yes	Yes
avg_occupancy	Yes	Yes	Yes	Yes
avg_speed	Yes	Yes	Yes	Yes

³ Requires Bi Tran 233RV2 firmware and adequate field communications.

⁴ To be added to the IEN TCS Interface at RIITS's request.