



## SEARCH

The National Consortium for Justice Information and Statistics

## Using NIEM with Web Services

By Andrew Owen

Justice Information Systems Specialist  
SEARCH

### Purpose

The [National Information Exchange Model](#) (NIEM) is an XML vocabulary that is quickly gaining popularity as a means to support cross-domain information sharing. NIEM is about the *semantics* of the message, not *transportation* of that message. “Web services” is a term for a group of industry standards that collectively provide a mechanism for exchanging XML-based messages, such as NIEM messages. One of these standards is the Web Services Description Language (WSDL), which standardizes the specification and description of a web services interface. This technical brief explains the crucial relationship between NIEM Information Exchange Package Documentation (IEPD) and WSDL.

### Intended Audience and Usage

Individuals experienced with NIEM, IEPDs, and web services (especially WSDL) should use this brief to gain a better understanding of how WSDL can use NIEM IEPDs as a way to standardize content of a web service.

### Brief Refresher

#### *NIEM and IEPDs*

NIEM is a large and extensible XML data model composed of thousands of data components and associations among those components. NIEM supports information sharing within and across multiple domains through developing IEPDs to standardize on the semantics of a message. An IEPD, or exchange specification, is a collection of mutually supportive artifacts (including XML schema) that define the content of a specific information exchange.

**NIEM:** An XML vocabulary that supports cross-domain information sharing by standardizing the semantics of a message.

**Web services:** A group of industry standards that provide a mechanism for exchanging XML-based messages.

**WSDL:** Standardizes the specification and description of a web services interface.

**IEPD:** A collection of mutually supportive artifacts that define the content of a specific information exchange.

## ***Web Services***

A web service is an open standards-based software component that supports machine-to-machine interaction via messages. XML is the basis for web services-based communication. Important to this brief is the concept of WSDL. Web services can be explicitly described by WSDL—the de facto standard language for defining web services interfaces.

An appealing characteristic of web services is that they are self-describing. What this means is that a WSDL can contain the necessary information for service consumers to interact with a web service. Through use of WSDL and IEPD together, certain tools can generate endpoint code that developers can, in turn, use to facilitate the process of sending and receiving messages.

## **IEPD and WSDL**

The information that travels between systems is critical to the workings of a web service. A WSDL has the responsibility of specifically defining what that information looks like. Though IEPD is a concept separate from web services, an IEPD can be used in conjunction with WSDL to clearly and unambiguously specify a web services interface, including all the semantics. In other words, web services can be used for exchanging messages defined by NIEM IEPDs.

WSDL relies on seven major elements or structures to define a web service. This document is mainly concerned with three of those structures, called “types,” “message,” and “operation”:

- The **types** structure contains data definitions used to describe the messages exchanged as part of the service.<sup>1</sup>
- The **message** structure represents an abstract definition of transmitted data; each “message” consists of one or more logical “parts.”<sup>2</sup> Each “message part” must be defined by an element or type defined in XML schema.
- The **operation** structure contains an abstract description of an action supported by the service.<sup>3</sup>

---

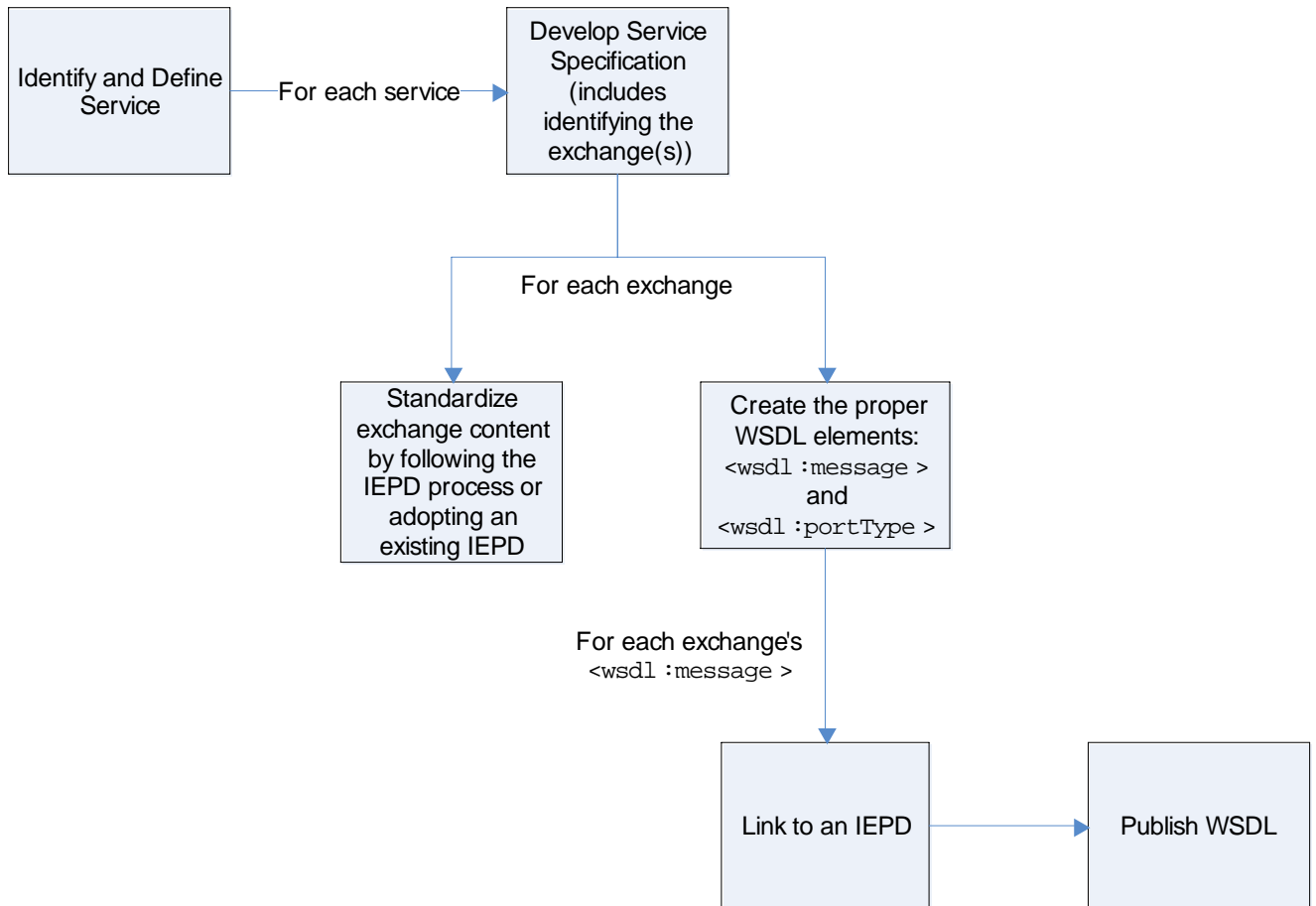
<sup>1</sup> As defined by Web Services Description Language (WSDL) 1.1 <http://www.w3.org/TR/wsdl>.

<sup>2</sup> Ibid.

<sup>3</sup> Ibid.

## Defining a Web Service

This section describes the general steps required to defining web services.



- **Identify and Define Service**<sup>4</sup> – Determine and prioritize capabilities that the business owns that may be valuable to other businesses. These capabilities can be exposed through services.
- **Develop the Service Specification**<sup>5</sup> – For each service, document the capabilities made available through the service; the service model (behavioral model, information model, and interactions); the policies that constrain the use of the service; and the service interface that provides a means of interaction with the service.

<sup>4</sup> The [Global Infrastructure/Standards Working Group](#) (GISWG) is currently developing formal guidance around identifying and defining services. GISWG will make this documentation available in mid-2009.

<sup>5</sup> GISWG is currently developing formal guidance around developing service specifications. GISWG will make this documentation available in mid-2009.

- **Standardize Exchange Content**<sup>6</sup> – For each exchange that can be performed against the service, follow the IEPD process to standardize the content of that exchange. This could involve developing a new IEPD and/or reusing an existing IEPD.
- **Create the Proper WSDL Elements**<sup>7</sup> – for each exchange, create the proper `<wsdl:message>` and `<wsdl:portType>` elements in the WSDL. The structure of `<wsdl:portType>` will correspond to the Message Exchange Pattern (MEP)<sup>8</sup> identified in the Service Specification.
- **Link to an IEPD**<sup>9</sup> – For each `<wsdl:message>`, link it to the corresponding root element in an IEPD.
- **Publish WSDL**<sup>10</sup> – Make the WSDL specification available to potential service consumers so they're able to design their systems to interact with the web service.

### *WSDL Example*

This section contains an example of a WSDL that defines an Arrest Report Service. This web service relies on a NIEM IEPD that defines the content of an Arrest Report. This document doesn't specifically show the IEPD artifacts, though we are able to make certain assumptions about the IEPD based on information in the WSDL.

---

<sup>6</sup> Visit <http://www.niem.gov> for more information about using NIEM to standardize information exchanges.

<sup>7</sup> Visit <http://www.w3.org/TR/wsdl> to learn more about WSDL and its components.

<sup>8</sup> Download [http://www.it.ojp.gov/documents/JRA\\_Specification\\_1-7.doc](http://www.it.ojp.gov/documents/JRA_Specification_1-7.doc) and [http://www.it.ojp.gov/documents/WS-SIP\\_Aug\\_31\\_version\\_1\\_1\\_FINAL\(3\).pdf](http://www.it.ojp.gov/documents/WS-SIP_Aug_31_version_1_1_FINAL(3).pdf) to learn more about MEPs.

<sup>9</sup> Visit [http://www.w3.org/TR/wsdl#\\_messages](http://www.w3.org/TR/wsdl#_messages) to learn more about defining content of a WSDL message.

<sup>10</sup> WSDL is part of a broader package of artifacts that includes conceptual descriptions and logical models of the service. A WSDL would be part of the formal Service Specification.

```

<definitions
  targetNamespace="http://local.gov/ws/Arrest"
  xmlns:bws="http://local.gov/ws/Arrest"
  xmlns:bd="http://local.gov/exchange"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">

  <types>
    <xsd:schema>
      <xsd:import namespace="http://local.gov/exchange"
schemaLocation="xsd/ext/exchange.xsd"/>
    </xsd:schema>
  </types>

  <message name="ArrestReportRequest">
    <part name="ArrestReportMessage" element="bd:ArrestReport"/>
  </message>

  <message name="ArrestReportResponse">
    <part name="ArrestReportResponseMessage"
element="bd:ResponseMessage"/>
  </message>

  <portType name="ArrestReportPort">
    <operation name="ReportNewArrest">
      <input name="ArrestReportRequest"
message="bws:ArrestReportRequest"/>
      <output name="ArrestReportResponse"
message="bws:ArrestReportResponse"/>
    </operation>
  </portType>
  <binding name="ArrestReportPortSOAPBinding" type="bws:ArrestReportPort">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="ReportNewArrest">
      <soap:operation soapAction="ReportNewArrest"
style="document"/>
      <input name="ArrestReportRequest">
        <soap:body use="literal"/>
      </input>
      <output name="ArrestReportResponse">
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="ArrestReportService">
    <port name="ArrestReportPort"
binding="bws:ArrestReportPortSOAPBinding">
      <soap:address
location="https://localhost:0000/soap/ArrestReportService"/>
    </port>
  </service>
</definitions>

```

## Types

The blue text in the WSDL file is the **types** structure. As mentioned earlier in this document, the types structure contains data definitions that can be used to describe the messages exchanged as part of the service. Essentially, the types structure can define its own XML schema(s) or import an existing external XML schema(s). In this example, the types structure contains an import of a schema that happens to be the exchange schema for the Arrest Report IEPD. By importing the exchange schema, the WSDL now has access to the Arrest Report structure as defined by the Arrest Report IEPD.

## Message

The red text in the WSDL file shows two different **message** structures. As mentioned earlier, a message structure represents an abstract definition of transmitted data. Each message is divided into one or more parts, each of which is defined by an XML element or type. The first message, “ArrestReportRequest,” has a structure as defined by the “ArrestReport” element in the `http://local.gov/exchange` namespace. The second message, “ArrestReportResponse,” has a structure as defined by “ResponseMessage” element in the `http://local.gov/exchange` namespace. These messages can be used later in the WSDL to define the inputs and/or outputs of operations.

The namespace that we imported in the previous section is `http://local.gov/exchange` namespace; this namespace has been defined as part of the Arrest Report IEPD.

## Operation

The green text in the WSDL file shows the **operation** structure. As mentioned earlier, the operation structure contains an abstract description of an action supported by the service. More specifically, the operation structure describes the inputs and outputs of an action against the service. The input and output of an action are specified by WSDL messages, which we know can be defined by NIEM IEPD content. The WSDL example contains a single operation called “ReportNewArrest.” The input to this operation is defined by “ArrestReportRequestMessage” and the output of this operation is defined by “ArrestReportResponseMessage.”

## Conclusion

Important to web services is the standardization of data that flows into and out of the service. A web service does not provide its own language for standardizing that content. Instead, domain vocabularies like NIEM must be used. When leveraged properly, NIEM IEPDs can standardize the data that is to be consumed and/or produced by a web service.

The [Global Information Sharing Initiative](#) (Global), through the [Global Infrastructure/Standards Working Group](#), provides specific guidance on use of web services and NIEM IEPDs to support Service Oriented Architecture (SOA). This guidance is provided in the form of the Justice Reference Architecture (JRA) web services service interaction profile (SIP). The web services SIP is publicly available at <http://www.it.ojp.gov/globaljra>.

This project was supported by funding from the U.S. Department of Justice Bureau of Justice Assistance. Points of view or opinions contained in this document are those of the author and do not necessarily represent the official position or policies of the U.S. Department of Justice.

**Francis X. Aumand II**  
Chairman

**Ronald P. Hawley**  
Executive Director

**Kelly J. Peters**  
Deputy Executive Director

### **SEARCH**

7311 Greenhaven Drive, Suite 145 • Sacramento, CA 95831  
(916) 392-2550 • (916) 392-8440 (fax) • [www.search.org](http://www.search.org)