

BIAS WSDL Review :: 2010-10-12

National Institute of Standards and Technology
Information Technology Lab / Information Access Division / Image Group
Biometrics Clients Lab
Prepared by Kevin Mangold

Contents

Initial Notes.....	4
Platform	4
Validation and Code Generation tools	4
XSV.....	4
Information	4
Input	4
Output.....	4
Eclipse IDE for Java EE Developers (with Web Developer Tools)	5
Information	5
Input	5
Output.....	6
Java SE 1.6.0 SDK (JAX-WS)	6
Information	6
Input	6
Output.....	6
Apache Axis 2	7
Information	7
Input	7
Output.....	7
Microsoft Visual Studio 2010 Premium.....	7
Information	7
Input	7
Output.....	7
IBM Schema Quality	7
Information	7
Input	8

Output.....	8
Identified Issues	8
Abstract Complex Types.....	8
Problem.....	8
Solution.....	8
Schema Sequence has Unbounded Maximum Occurrences	8
Problem.....	8
Solution.....	8
Schema Attributes	8
Problem.....	8
Solution.....	9
Schema Choice Element.....	9
Problem.....	9
Solution.....	9
Request/Response Element Naming.....	9
Problem.....	9
Solution.....	9
Nilable Elements.....	9
Problem.....	9
Solution.....	9
Request/Response Element Inheritance.....	10
Problem.....	10
Solution.....	10
WSDL Message Part Name.....	10
Problem.....	10
Solution.....	10
Code Generation with xsd:any.....	10
Problem.....	10
Concluding Notes	10
References.....	11
Appendix A: Updated WSDL	11

Initial Notes

The purpose of the following updates is to promote the adoption of the BIAS WSDL through usability. The ISO 9241-11 standard defines usability as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." Software developers commonly use code generation tools to work more efficiently. If a developer is required to make changes to the schema, this takes a lot of time and will often invalidate conformance to the standard and break interoperability between separate systems. Relieving the need to make any changes to the schema increases the effectiveness of the schema and increases the efficiency at which a developer can integrate it with their own system. Learnability is another important characteristic of usability. Reinventing certain data types adds complexity to the schema and creates avoidable overhead in the learning process. Lowering the learning curve increases usability which, in turn, increases ease of adoption and integration. (Mangold, CBEFF Schema Updates, 2010)

Changes to the WSDL in this update have been made to conform to the specifications set out in the W3 Web Services Description Language (WSDL) 1.1 specification, WS-I Basic Profile 1.1 specification, ANSI INCITS 442-2008 Biometric Identity Assurances Services (BIAS), ISO 19785-3 Common Biometric Exchange Formats Framework (CBEFF), and OASIS Biometric Identity Assurance Services (BIAS) SOAP Profile 1.0. In addition, the tools used provided documentation and recommendations on how to best design the WSDL to allow for better code generation. (Mangold, BIAS WSDL Review 2010-08-11.pdf, 2010)

Platform

All of research is currently being performed on a PC with Windows Server 2008 R2 Standard x64 as the primary operating system.

Validation and Code Generation tools

As each different implementation of an XML/WSDL validation and/or code generation tool handles different elements/attributes differently, I decided to use multiple widely tools to cover a wider range of inconsistencies.

XSV

Information

- Version: 3.1-1
- URL: <http://www.w3.org/2001/03/webdata/xsv>

XSV is an XML Schema Validation tool, with a web front-end and a standalone executable, supported by the W3 organization.

Input

```
xsv BIAS.wsdl
```

Output

```
<?xml version='1.0'?>
<xsv xmlns=http://www.w3.org/2000/05/xsv
      docElt="{http://schemas.xmlsoap.org/wsdl/}definitions"
```

```

instanceAssessed="true" instanceErrors="0"
rootType="{http://schemas.xmlsoap.org/wsdl/}:tDefinitions"
schemaErrors="0"
target=file:///C:/BIAS/WSDL/BIAS.wsdl
validation="strict" version="XSV 3.1-1 of 2007/12/11 16:20:05">
<schemaDocAttempt URI=http://schemas.xmlsoap.org/wsdl/
namespace=http://schemas.xmlsoap.org/wsdl/
outcome="success" source="docElt"/>
<schemaDocAttempt URI=http://schemas.xmlsoap.org/wsdl/soap/
namespace=http://schemas.xmlsoap.org/wsdl/soap/
outcome="success" source="new namespace"/>
<schemaDocAttempt URI=http://schemas.xmlsoap.org/wsdl/
namespace=http://schemas.xmlsoap.org/wsdl/
outcome="redundant" source="import"/>
</xsv>

```

Eclipse IDE for Java EE Developers (with Web Developer Tools)

Information

- IDE Version: Eclipse Helios Release (1.3.0.20100617-0521)
- Web Developer Tools Version 3.2.0.v201005241510
- URL: <http://www.eclipse.org/downloads/moreinfo/jee.php>

The Eclipse IDE is a very popular integrated development environment (IDE) for developing in the Java programming language. It has been extended to support XML, Schema, WSDL, and web service development in the Java EE version of the IDE.

Input

File -> Open File

Navigate and select the WSDL file.

Click "Open"

Output

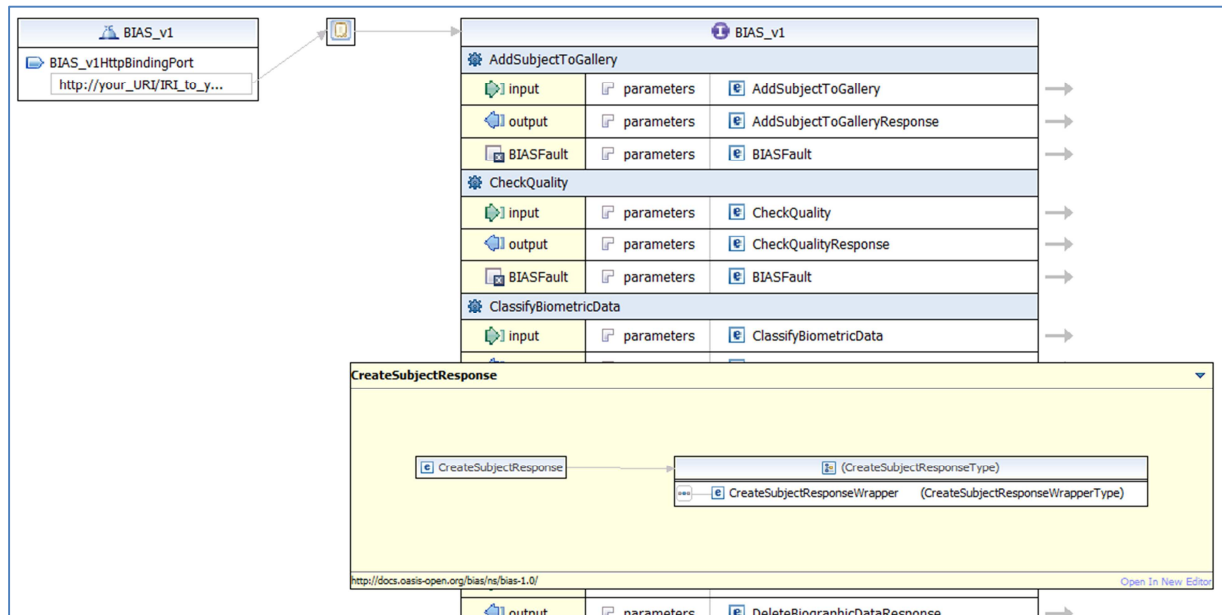


Figure 1 - Partial screenshot of the Eclipse visual representation of the modified WSDL

Java SE 1.6.0 SDK (JAX-WS)

Information

- o Java Version: 1.6.0_21
- o JAX-WS Version: 2.1.6
- o URL: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

The Java 1.6.0 SDK comes with its own code generation tools relating to web services. The tool used for generating Java code from a WSDL is called "wsimport" which reads and parses a WSDL file and generates Java classes which can be used when sending and receiving web service messages.

Input

```
wsimport -d %CD%\wsimport BIAS.wsdl
```

Output

```
parsing WSDL...
```

```
generating code...
```

```
compiling code...
```

Note:

```
C:\BIAS\WSDL\wsimport\org\oasis_open\docs\bias\ns\bias_1\ObjectFactory
.java uses unchecked or unsafe operations.
```

Note: Recompile with -Xlint:unchecked for details.

Apache Axis 2

Information

- Version: 1.5.1
- URL: <http://ws.apache.org/axis2/>

Apache Axis 2 is a popular open source Web Service/WSDL/SOAP engine developed and maintained by the Apache development community. The tool used, wsdl2java, does precisely what its name says: takes a WSDL file and creates Java code from it.

Input

```
wsdl2java -o %CD%\axis2 -uri %CD%\BIAS.wsdl
```

Output

```
Using AXIS2_HOME:    C:\axis2-1.5.1
Using JAVA_HOME:     C:\Program Files\Java\jdk1.6.0_21
Retrieving document at 'C:\BIAS\WSDL\BIAS.wsdl'.
Retrieving schema at 'cbeff.xsd', relative to
'file:/C:/BIAS/WSDL/BIAS.wsdl'.
```

Microsoft Visual Studio 2010 Premium

Information

- Version: 10.0.30319.1
- .NET Framework Version: 4.0.30319
- URL: <http://www.microsoft.com/visualstudio/en-us/products/2010-editions/premium>

Microsoft Visual Studio (MSVS) 2010 Premium is an integrated development environment (IDE) for VB.NET, C#, and other .NET programming languages. MSVS does have a command line tool for generating code from a WSDL file; however, to keep things simple for now and to provide different scenarios, I kept with the GUI implementation.

Input

File -> Open -> File...

Navigate and select the WSDL file.

Click "Open"

Output

No errors or warnings.

IBM Schema Quality

Information

- Version: 2.2.1
- URL: <http://www.alphaworks.ibm.com/tech/xmlsqc>

IBM Schema Quality is similar to XSV, in that it verifies that a schema file has been correctly written. This tool also has a `-wsdl` flag which allows a WSDL file to be processed and validated.

Input

```
SQC.bat -wsdl \BIAS\WSDL\BIAS.wsdl
```

Output

See "TC_20101012_SQC_OUTPUT.txt"



TC_20101012_SQC_OUTPUT.txt

Identified Issues

The following issues are those that were mentioned in the previous BIAS WSDL Review submission on August 11, 2010 as well as issues that have been discovered since then.

Abstract Complex Types

Problem

Marking a schema type as abstract only provides the assurance that the specific type will not appear unless it is inherited. There is nothing stopping a developer from inheriting that type and not extending it any further, essentially finding a way around the abstract restriction. Additionally, for a similar reason that you can't require subtypes to add more child elements, code generation tools often ignore or fail when it comes across the abstract attribute.

Solution

Remove the abstract attributes from all types defined in the schema.

Schema Sequence has Unbounded Maximum Occurrences

Problem

When the original schema is used in conjunction with common code generation tools, the output is obfuscated and messy; it does not generate logically designed code.

Solution

The solution to this problem is, first, to remove the unbounded maximum occurrences attribute. If the sequence contains one element, set that element's maximum occurrences to unbounded. If there are two or more elements, put these elements inside a wrapper type/element; now the sequence only contains one element and can fall back to the previous solution where a single element's maximum occurrences attribute can be set to unbounded.

Schema Attributes

Problem

Once a code generation tool parses a schema document, it converts all attributes and elements into class variables, essentially treating both attributes and elements the same. Because of this, some code generation

tools do not have support for attributes. In other words, any time a schema attribute is defined, the tool will either ignore it or fail to properly parse the schema.

Solution

Convert all schema attributes into schema elements. Switching attributes into elements does not add or remove any functionality of the schema; however, it enhances its utility when put through common developer tools.

Schema Choice Element

Problem

In Schema, using a choice element seems like the way to go when there are multiple elements that could appear in a given location. However, in object-oriented programming languages, the most logical way to represent this is to have a base class or interface and subclasses or implementing classes. There is no directly-mappable way to go from choice to inheritance and vice-versa. Because of this, code generation tools often ignore, fail, or leave it up to the developer to manually parse and traverse the XML document model, when it comes across a choice element.

Solution

Replace the schema's choice element with a sequence element containing one child. This child element will act as a base class. The elements previously listed inside the choice element will be first moved out to become its own type then updated to inherit from the particular base class.

Request/Response Element Naming

Problem

The most common convention when dealing with WSDL documents is called the document/literal wrapped convention. Most code generation tools and other WSDL utility tools handle this, at a minimum, because of its widespread use. To support this convention, elements linked to `wsdl:messages/wsdl:operations` must have a specific name. There must be two elements. One element, the input element, must have the same name as the operation. The second element, the output element, must have the same name as the operation appended by "Response". The original schema does not support this naming convention.

Solution

Change the names of all elements directly related to a `wsdl:message/wsdl:operation`. The standard convention regarding this is that the request/input element is to be named the same as the operation name. The response/output element is to be named the same as the operation name appended by "Response".

Nilable Elements

Problem

The most common convention when dealing with WSDL documents is called the document/literal wrapped convention. Most code generation tools and other WSDL utility tools handle this, at a minimum, because of its widespread use. To support this convention, elements linked to the input/output `wsdl:messages` must have the `nilable` attribute set to true on all direct child elements. This is not true for the original schema.

Solution

Add a `nilable` attribute to each element directly mapped to a `wsdl:message` and set it as true.

Request/Response Element Inheritance

Problem

WSDL documents are constructed in a way that there are sets of operations. Each operation has an input and output message and, optionally, a fault message. Each message is bound to an element defined in the schema. Code generation tools often assume, implicitly or not, that it is constructed in a specific way. One of the more common assumptions is that the elements that are bound to each message do not directly inherit from another type. Violating this assumption causes the tool to fail or not properly generate classes that hold the message structure.

Solution

Wrap the contents of this element with another element that does not inherit from anything explicitly.

WSDL Message Part Name

Problem

The most common convention when dealing with WSDL documents is called the document/literal wrapped convention. Most code generation tools and other WSDL utility tools handle this, at a minimum, because of its widespread use. To support this convention, each `wsdl:message` element has one child element, `wsdl:parameter`. The name attribute of all `wsdl:part` elements must be "parameters". This is not true for the original schema.

Solution

The `wsdl:message` element has a child element, `wsdl:part`; change the `wsdl:part`'s name attribute to "parameters" for all instances.

Code Generation with `xsd:any`

Problem

Adding wildcards, such as the `xsd:any` type, to an XML schema may be a good idea as it allows a loose coupling between implementation-specific instances, it is not fully supported by many WSDL to Code generation tools. The extent that most of the popular tools support the *any* type is that it will generate a class member of the type: *XmlElement*, *SOAPElement*, or the language-specific class name. This means that the developers writing up against this WSDL/API will need to manually create an XML tree structure from within the code. This can be very tedious and painstaking, especially if the expected element is fairly complex in structure. Additionally, it is impossible to *extend* the *any* type for future versioning as there are no requirements on the base element. (Mangold, BIAS WSDL Review 2010-08-11.pdf, 2010)

Concluding Notes

Issues brought up in this document are not an absolute list. All issues mentioned, excluding a substitute for `xsd:any`, have recommended solutions. The necessity of a `xsd:any` substitute is still under review. NIST's reference implementation will be continued in an effort to validate the OASIS/BIAS standard as well as discover any additional recommendations to update the WSDL/schema.

References

- Ballinger, K., Ehnebuske, D., Ferris, C., Gudgin, M., Liu, C. K., Nottingham, M., et al. (2006, 04 10). *Basic Profile - Version 1.1 (Final)*. Retrieved from WS-I Organization: <http://www.ws-i.org/Profiles/BasicProfile-1.1.html>
- Butek, R. (2005, 12 13). *Tip: xsd:any: A cautionary tale*. Retrieved from IBM: <http://www.ibm.com/developerworks/xml/library/ws-tip-xsdcaution.html>
- Butek, R. (2005, 05 24). *Which style of WSDL should I use?* Retrieved from IBM: <http://www.ibm.com/developerworks/webservices/library/ws-whichwsdl/>
- Butek, R. (2008, 03 25). *Web services hints and tips: Design reusable WSDL faults*. Retrieved from IBM: <http://www.ibm.com/developerworks/webservices/library/ws-tip-wsdlfaults/index.html>
- Christensen, E., Curbera, F., Meredith, G., & Sanjiva, W. (2001, 03 15). *Web Service Definition Language (WSDL) 1.1*. Retrieved from World Wide Web Consortium: <http://www.w3.org/TR/wsdl>
- Herath, N. M. (2008, 12 6). *How to Convert RPC/encoded WSDL into a Document/literal Wrapped WSDL*. Retrieved from WSO2 - OxygenTank: The Developer Portal for SOA: <http://wso2.org/library/knowledge-base/convert-rpc-encoded-wsdl-document-literal-wrapped-wsdl>
- Icodeon. (2007, 07). *Building Document Literal Wrapped Web Services with Java*. Retrieved from Icodeon: <http://labs.icodeon.com/projects/spd/html/index.html>
- Mangold, K. (2010, 08 11). *BIAS WSDL Review 2010-08-11.pdf*. Retrieved from <http://www.oasis-open.org/committees/download.php/38923/20100811.zip>.
- Mangold, K. (2010, 09 30). CBEFF Schema Updates.
- Nottingham, M. (2004, 08 24). *Simple SOAP Binding Profile - Version 1.0 (Final)*. Retrieved from WS-I Organization: <http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0.html>

Appendix A: Updated WSDL

See "BIAS.wsdl"



BIAS.wsdl