



MedBiquitous Web Services Design Guidelines

Version 2.0

**13 May 2009
MedBiquitous Technical Steering Committee**

MedBiquitous	Version: 2.0
Web Services Design Guidelines	Date: 13 May 2009

Revision History

Date	Version	Description	Author
17 Dec 2003	0.9	Draft for Technical Steering Committee review	Scott Hinkelman srh@us.ibm.com
17 Dec 2003	0.96	Namespace updates	Scott Hinkelman srh@us.ibm.com
30 Apr 2004	1.0	Updated format	Valerie Smothers valerie.smothers@medbiq.org
13 May 2009	2.0	References WSI Basic profile 1.1, replacing 1.1 References SOAP Message Transmission Optimization Mechanism replacing SOAP with attachments References RFC 3986 for URIs Recommends WS-Security to pass credentials within the soap request and references WSI Basic Security Profile 1.0. Eliminate common fault messages. Reference WS-I Basic Profile 1.2 for MTOM implementations	Valerie Smothers Valerie.smothers@medbiq.org

MedBiquitous	Version: 2.0
Web Services Design Guidelines	Date: 13 May 2009

MedBiquitous Consortium XML Public License and Terms of Use

MedBiquitous XML (including schemas, specifications, sample documents, Web services description files, and related items) is provided by the copyright holders under the following license. By obtaining, using, and or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

The Consortium hereby grants a perpetual, non-exclusive, non-transferable, license to copy, use, display, perform, modify, make derivative works of, and develop the MedBiquitous XML for any use and without any fee or royalty, provided that you include the following on ALL copies of the MedBiquitous XML or portions thereof, including modifications, that you make.

1. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the following notice should be used: "Copyright © [date of XML release] MedBiquitous Consortium. All Rights Reserved. <http://www.medbiq.org>"
2. Notice of any changes or modification to the MedBiquitous XML files.
3. Notice that any user is bound by the terms of this license and reference to the full text of this license in a location viewable to users of the redistributed or derivative work.

In the event that the licensee modifies any part of the MedBiquitous XML, it will not then represent to the public, through any act or omission, that the resulting modification is an official specification of the MedBiquitous Consortium unless and until such modification is officially adopted.

THE CONSORTIUM MAKES NO WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, WITH RESPECT TO ANY COMPUTER CODE, INCLUDING SCHEMAS, SPECIFICATIONS, SAMPLE DOCUMENTS, WEB SERVICES DESCRIPTION FILES, AND RELATED ITEMS. WITHOUT LIMITING THE FOREGOING, THE CONSORTIUM DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY, EXPRESS OR IMPLIED, AGAINST INFRINGEMENT BY THE MEDBIQUITOUS XML OF ANY THIRD PARTY PATENTS, TRADEMARKS, COPYRIGHTS OR OTHER RIGHTS. THE LICENSEE AGREES THAT ALL COMPUTER CODES OR RELATED ITEMS PROVIDED SHALL BE ACCEPTED BY LICENSEE "AS IS". THUS, THE ENTIRE RISK OF NON-PERFORMANCE OF THE MEDBIQUITOUS XML RESTS WITH THE LICENSEE WHO SHALL BEAR ALL COSTS OF ANY SERVICE, REPAIR OR CORRECTION.

IN NO EVENT SHALL THE CONSORTIUM OR ITS MEMBERS BE LIABLE TO THE LICENSEE OR ANY OTHER USER FOR DAMAGES OF ANY NATURE, INCLUDING, WITHOUT LIMITATION, ANY GENERAL, DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, OR SPECIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF ANY USE OF MEDBIQUITOUS XML.

LICENSEE SHALL INDEMNIFY THE CONSORTIUM AND EACH OF ITS MEMBERS FROM ANY LOSS, CLAIM, DAMAGE OR LIABILITY (INCLUDING, WITHOUT LIMITATION, PAYMENT OF ATTORNEYS' FEES AND COURT COSTS) ARISING OUT OF MODIFICATION OR USE OF THE MEDBIQUITOUS XML OR ANY RELATED CONTENT OR MATERIAL BY LICENSEE.

LICENSEE SHALL NOT OBTAIN OR ATTEMPT TO OBTAIN ANY PATENTS, COPYRIGHTS OR OTHER PROPRIETARY RIGHTS WITH RESPECT TO THE MEDBIQUITOUS XML.

THIS LICENSE SHALL TERMINATE AUTOMATICALLY IF LICENSEE VIOLATES ANY OF ITS TERMS AND CONDITIONS.

MedBiquitous	Version: 2.0
Web Services Design Guidelines	Date: 13 May 2009

Table of Contents

MedBiquitous Consortium XML Public License and Terms of Use	3
1. Acknowledgements	5
2. Scope	5
3. Status	5
4. Introduction	5
5. Assumptions & Goals	5
5.1 Assumptions	5
5.2 Goals	5
6. Notational Conventions	5
7. Service Versioning and Namespace	6
8. Service Filenames and Location	6
9. Business Information Interfaces and Authoring	6
9.1 Payload Authoring	6
9.2 Intentionality using WSDL Operation Names	7
9.3 Operation Granularity	7
9.4 Interface Statefulness	7
10. Uses of Current Web Services Technology	7
10.1 Transport	7
10.2 Point to Point Security	7
10.3 Protocol and Wire Format Binding	7
10.4 Service Definition and Authoring	9
10.4.1 'What' WSDL Perspective	10
10.5 Business Process	11
10.6 Discovery	11
10.7 Binary Information Transmit	11
11. References	11

MedBiquitous	Version: 2.0
Web Services Design Guidelines	Date: 13 May 2009

Web Services Design Guidelines

1. Acknowledgements

These guidelines are based on a submission from Scott Hinkelman of IBM. Joel Farrell of IBM, Darin McBeath of Elsevier Science, Dan Rehak, and Carl Singer of CECity also contributed to this document.

2. Scope

This document suggests best practices, high level guidelines, and recommendations that should be followed when defining MedBiquitous Web services. It defines the approach for concrete realization within a Web services environment.

3. Status

This document is considered part of MedBiquitous release 1.0. MedBiquitous welcomes your suggestions about this document.

4. Introduction

MedBiquitous is committed to defining a Web services-based environment to support the exchange of MedBiquitous payload information. This document is focused on recommendations and guidelines for Web services design based on current Web services specifications/standards. It is expected that the next release of this document will address increased emerging technology and directions. It should be realized that the content of this document is closely tied to the date/version due to the evolving Web services technology, and revisions will be required for the foreseeable future.

5. Assumptions & Goals

5.1 Assumptions

- Readers have an understanding of the base de facto Web Services technologies.
- Readers have an understanding of the state of the industry with regard to the current stability and continual evolution of such technology.
- Readers have an understanding of interface definition paradigms and the role in distributed computing.

5.2 Goals

The primary goal of this document is to provide the initial guidelines for development and implementation of MedBiquitous Web services. This document expresses guidelines for consistency based in real world experience for the purpose of initially moving forward with Web services.

It should be noted that WS-I.org's basic profile version 1.1 [[WSIBP](#)] provides a wealth of guidelines focused on achieving interoperability and should be considered along with this document. However, some of WS-I's basic profile is a clarification of Web service specifications targeted for Web services infrastructure developers and is not directly relevant to MedBiquitous Web Services guidelines. Also, there are sections within the WS-I basic profile that identify inconsistencies between the WSDL1.1 [[WSDL](#)] specification and its W3C XML Schema [[XMLSchema1](#)], which are not referenced within this document.

Where relevant, the WS-I information is referenced as appropriate, but the WS-I BP 1.1 should be considered required reading. This document goes beyond the space of WS-I and addresses guidelines for items such as common faults for all services, which are needed to accommodate a robust Web services approach while considering issues such as the payload design approach, etc.

6. Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in

MedBiquitous	Version: 2.0
Web Services Design Guidelines	Date: 13 May 2009

RFC2119 [RFC2119]. Namespace URIs (of the general form "some-URI") represents some application-dependent or context-dependent URI [URI] as defined in [RFC3986].

7. Service Versioning and Namespace

The MedBiquitous namespace URIs are of a URL scheme. As such, nothing is guaranteed about re-referencing the namespace URL at runtime, and the schema location is given by the schemaLocation attribute of xsd:import element.

MedBiquitous uses the namespace name to represent versioning of the business content, with an initial version of **1**. Any structural or syntactic changes to the XML Schema file requires the version number to change.

Example of a member workgroup namespace name:

```
http://ns.medbiq.org/member/v1/
```

The namespace name representing MedBiquitous Web services will be formulated by a consistent modification of the MedBiquitous business content namespace scheme, as defined in the MedBiquitous XML Schema Design Guidelines 1.0. A service namespace name will be the same as the schemas for the business content, with **service** inserted before the version number.

An example of the matching service namespace:

```
http://ns.medbiq.org/member/service/v1/
```

8. Service Filenames and Location

The MedBiquitous service binding convention utilizes a wrapped document-literal approach, a business-content wrapped model (see protocol section), where the business content schemas defined by the business working groups are imported by the schema defining the service operations.

The MedBiquitous wrapper schemas will be defined within the associated service namespace.

The wrapper file names will follow the pattern (workgroup)types.xsd.

Example of a member workgroup namespace name:

```
membertypes.xsd
```

The associated files would available in a **v1** directory which will have an index.html document that contains links to the schemas and to any specifications. The associated files would be

```
http://ns.medbiq.org/member/v1/member.xsd
http://ns.medbiq.org/member/service/v1/membertypes.xsd
```

9. Business Information Interfaces and Authoring

9.1 Payload Authoring

MedBiquitous will avoid defining business content in the <wsdl:type> structure and keep all business type information authored independently from the Web services interface specification documents. This will help enable the business payload development by avoiding the details of the underlying Web services technology layer. MedBiquitous payload xsd's will be imported indirectly into MedBiquitous WSDL files through the (workgroup)types.xsd files. Business content will not be authored within WSDL files.

MedBiquitous	Version: 2.0
Web Services Design Guidelines	Date: 13 May 2009

9.2 Intentionality using WSDL Operation Names

The current MedBiquitous business content design approach is to design XML documents within a project scope without providing any information on how the documents will be used (non-contextual information modeling). For example, the Member Profile schema provides a method of describing a member record but excludes information on what is to be done with the member record. WSDL1.1's abstract layer allows Web services designers to express how the documents will be used through named operations. Operations provide the usage context by expressing the intentionality (verbs). This is the predominant situation in MedBiquitous. MedBiquitous will use WSDL operation names to express intentionality in this case, with the operations names used as the root element of input messages defined within the (workgroup)types.xsd files.

9.3 Operation Granularity

In order to avoid fragile Web service interfaces, consideration should be given to defining general operations on service interfaces before defining highly specific operations. An example of this is to prefer CRUD (Create, Read, Update, Delete) operations over operations such as "createCardiologist()" if appropriate. The preferred/common payload design authoring in MedBiquitous is to specify a robust type-base document using a single root element (Venetian Blind), which can be used for several operations.

This design lends towards WSDL1.1 <part> elements which reference an entire document via the wrapped element within the (workgroup)types.xsd file. This approach results in the WSDL <message> having a single <part>, leading to a natural approach toward generic interface operations across the entire multi-usage payload. This shields the interface from change over time, however at the expense of constraint checking within the business logic. To help mitigate the impact on business logic for these type of interfaces, MedBiquitous will define consistent 'business rule' faults for all WSDL <operations> (discussed further in this document). MedBiquitous business interface designs will favor generic service operations before highly typed specific operations.

9.4 Interface Statefulness

The current state of the Web services art is that there is no inherent architectural state mechanism. Several underlying transports provide state management capabilities, but there is no formalism within context of the Web services principals of separation of description and transport (bindings). MedBiquitous will design business interfaces with no reliance on statefulness.

10. Uses of Current Web Services Technology

10.1 Transport

There is only one formal transport binding (HTTP) defined for Web services assuming the SOAP protocol. For maximum interoperability, MedBiquitous will use HTTP for transport. This is consistent with [WS-I BP 1.1](#). MedBiquitous will use HTTP/1.0/1.1 for transport binding.

10.2 Point to Point Security

Initial implementations of MedBiquitous Web services will be largely point-to-point. For these environments, SSL can be used for mutual or one-way authentication and for data encryption support for confidentiality between two points. In environments beyond point-to-point, proof of an original subject's authentication and authorization credentials will not propagate across points/nodes and other Web services will need to be used to accomplish security goals. If parties to web service interaction require credentials to be passed in soap message, use WS-Security [\[WS-Security\]](#) as outlined in WS-I basic security profile 1.0 [\[WS-I Basic Security\]](#).

10.3 Protocol and Wire Format Binding

There is only one formal protocol binding for WSDL1.1 (SOAP1.1). MedBiquitous will utilize this protocol binding. MedBiquitous will use SOAP1.1 for interaction protocol.

WS-I.org has identified problems with several binding usages concerning the soapbind namespace attribute. MedBiquitous is concerned with the document/literal usage, where the namespace attribute should not be specified. This is consistent with the document/literal discussion in WS-I BP 1.1

MedBiquitous	Version: 2.0
Web Services Design Guidelines	Date: 13 May 2009

A wsdl:binding that contains a soapbind:binding element whose style attribute is either missing or specified with a value of **document** and whose set of soapbind:body elements all specify the **use="literal"** attribute constitute a document/literal binding. MedBiquitous' current direction defining type-oriented payload documents with single root elements lends itself toward defining WSDL document/literal bindings (RPC/literal bindings for example are restricted to referring to wsdl:part elements that have been defined using the type attribute). However, conclusions within WS-I.org are largely interpreted to indicate that payload uniqueness is required in order to distinguish payload intention at runtime. In order to facilitate this, and continue to allow the MedBiquitous workgroups to not be concerned with the low level infrastructure of Web services, MedBiquitous will deploy a 'wrapped' Document/Literal style, where the MedBiquitous content payloads are wrapped with request/response operation wrapper elements defined in a (workgroup)types.xsd file.

For the following examples, a fictitious MedBiquitous workgroup is used called 'What.' The following example typifies a MedBiquitous business content payload schema:

```
[ 1] <?xml version="1.0" encoding="UTF-8"?>
[ 2] <xs:schema targetNamespace="http://ns.medbiq.org/what/v1/"
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:w="http://ns.medbiq.org/what/v1/"
      elementFormDefault="qualified" attributeFormDefault="unqualified">
[ 3]   <xs:complexType name="WhatType">
[ 4]     <xs:sequence>
[ 5]       <xs:element name="A" maxOccurs="unbounded" />
[ 6]       <xs:element name="B" maxOccurs="unbounded" />
[ 7]     </xs:sequence>
[ 8]   </xs:complexType>
[ 9]   <xs:element name="Whats" type="w:WhatsType" />
[10]   <xs:complexType name="WhatsType">
[11]     <xs:sequence>
[12]       <xs:element name="What" type="w:WhatType"
maxOccurs="unbounded" />
[13]     </xs:sequence>
[14]   </xs:complexType>
[15] </xs:schema>
```

Lines (001) to (002) define the schema construct, namespace names, etc.

Lines (003) to (008) define a global complex type. Line (009) defines a single root element within the namespace of type defined on lines (010) to (014). Local elements are defined of a global type, such as on line (012).

The following is the matching whattypes.xsd wrapper schema:

```
[ 1] <?xml version="1.0" encoding="UTF-8"?>
[ 2] <xs:schema targetNamespace="ns.medbiq.org/what/service/v1/"
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:w="ns.medbiq.org/what/v1/"
      xmlns:ws="ns.medbiq.org/what/service/v1/"
      elementFormDefault="qualified" attributeFormDefault="unqualified">
[ 3]   <xs:import namespace="ns.medbiq.org/what/v1"
      schemaLocation="./what.xsd" />
[ 4]   <xs:element name="operation1">
[ 5]     <xs:complexType>
```

MedBiquitous	Version: 2.0
Web Services Design Guidelines	Date: 13 May 2009

```

[ 6]         <xs:sequence>
[ 7]             <xs:element ref="w:Whats" />
[ 8]         </xs:sequence>
[ 9]     </xs:complexType>
[10] </xs:element>
[11] <xs:element name="operation1Response">
[12]     <xs:complexType>
[13]         <xs:sequence>
[14]             <xs:element ref="w:Whats" />
[15]         </xs:sequence>
[16]     </xs:complexType>
[17] </xs:element>
[18] </xs:schema>

```

Lines (001) to (002) define the schema construct, namespaces, etc. The namespace name of the wrapper schema is the service namespace name on line (002).

Line (003) xs imports the what business content schema.

Lines (004) to (010) define a global element request wrapper for a WSDL operation specified with a name attribute of "Operation1". Line (007) defines a local element reference to the root business content element imported on line (003).

Lines (011) to (017) define a global element response wrapper for a WSDL operation specified with a name attribute of "Operation1". Line (014) defines a local element reference to the root business content element imported on line (003).

The operation wrapper elements will be in lower camel case to facilitate easy programmatic implementation. The operation wrapper elements will reside in a separate schema within the service namespace (see section on namespace/files names/ versioning). This will define the content structure needed for the MedBiquitous binding.

It should be realized that the wrapping element, reflective of each WSDL operation name, could be considered to essentially add definition to the payloads of MedBiquitous messages. Authoring of business content within the (workgroup)types.xsd files is restricted to defining structural associations of imported schema, with the primary intention to facilitate use of 3rd party schema which are a required part of the service but not appropriate to be integrated with the core MedBiquitous business content schemas. No first class business content will be introduced as part of the (workgroup)types.xsd files.

The wrapper schemas, one per service, will consist of the request and response elements for all operations defined within the service. The WSDL files will import this schema, and the core business content schema will be imported from the (workgroup)types.xsd file.

Concerning 3rd party schema locations, xsds for these schemas will be copied to the MedBiquitous schema server and then referenced by the schemaLocation attribute in order to remove dependency on 3rd party infrastructure.

10.4 Service Definition and Authoring

Web services define an IDL-based environment where the central gravity point is that of a service definition. It is well recognized with little contention that WSDL is the specification that will pervasively enable description of a 'service'. MedBiquitous must utilize WSDL1.1.

WSDL provides three logical perspectives on a Web service description:

- What (business interface issues)

MedBiquitous	Version: 2.0
Web Services Design Guidelines	Date: 13 May 2009

- How (binding issues)
- Where (service location issues)

10.4.1 'What' WSDL Perspective

MedBiquitous will define 'soap:binding' locations with only one associated Port. This is consistent with WSIBP 1.1.

Beyond specifying the binding as Doc/Lit within the how perspective of WSDL, the wrapping aspect largely falls into the 'what' perspective. The following example shows the 'what' WSDL perspective of the fictitious 'what' MedBiquitous workgroup consistently with the discussions in the protocol section of this document.

```
[1] <?xml version="1.0" encoding="UTF-8"?>
[2] <definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="ns.medbiq.org/what/service/v1"
  targetNamespace="ns.medbiq.org/what/service/v1"
  name="MedBiquitous What Service Example Version 1">
[3]   <types>
[4]     <xs:schema targetNamespace="ns.medbiq.org/what/service/v1">
[5]       <xs:include schemaLocation="./whattypes.xsd"/>
[6]     </xs:schema>
[7]   </types>
[8]   <message name="Operation1InputMessage">
[9]     <part name="Wrapper" element="tns:operation1"/>
[10]  </message>
[11]  <message name="Operation1OutputMessage">
[12]    <part name="Wrapper" element="tns:operation1Response"/>
[13]  </message>
[14]  <message name="Fault1Message">
[15]  </message>
[16]  <portType name="WhatPortType">
[17]    <operation name="Operation1">
[18]      <input message="tns:Operation1InputMessage"/>
[19]      <output message="tns:Operation1OutputMessage"/>
[20]      <fault name="Fault1" message="tns:Fault1Message"/>
[21]    </operation>
[22]  </portType>
[23] </definitions>
```

Lines (001) to (002) define the definitions, namespaces, etc. The target namespace name follows the consistent modification of the business content namespace name.

The WSDL types section from line (003) to (007) defines an W3C XML Schema in the service's target namespace name in order to xs include the whattypes.xsd wrapper schema on line (005), which is defined in the same namespace.

Lines (008) to (010) define an input message for the Operation1 operation. A single part, named 'Wrapper', is defined on line (009) which references the request global element for the operation in the included schema.

Lines (011) to (013) define an output message for the Operation1 operation. A single part, named 'Wrapper', is defined on line (012) which references the response global element for the operation in the included schema.

Line (014) defines a service-specific fault message.

MedBiquitous	Version: 2.0
Web Services Design Guidelines	Date: 13 May 2009

Lines (016) to (022) define the service portType. An operation is defined from line (017) to (021) which references the input and output messages on lines (018) and (019). Line (020) specifies the service-specific fault.

A full example of a Web service following these guidelines is available at: <http://ns.medbiq.org/member/service/v1/>

10.5 Business Process

MedBiquitous will ultimately move toward defining explicit business process definitions. However, it should be realized that simple stand-alone Web services invocations can be thought of as "implicit business processes." MedBiquitous can define/release initial sets of Web services, which can be used within such implicit processes without the initial focus on definition of "explicit processes" which use a formal declarative language. Currently there are several efforts in the business process space including the W3C Choreography workgroup, the Business Process Execution Language for Web Services in OASIS, and ebXML's BPSS. It is currently an area of immaturity and needs monitoring, but there is value to be currently gained by focusing on initial sets of service definitions outside of formal definitions for business process.

10.6 Discovery

Concerning registration mechanisms, formal tools for service descriptions, such as UDDI or ebXML RegRep, will not be included in the initial phases of MedBiquitous's Web services, as it has not been clear of the requirements (if any) for such. Decisions on utilizing a formal registry, when/if and hosting will be addressed later.

Publication and hosting of service definitions will be accommodated on the medbiq.org web site. A taxonomy will be defined by the TSC to accommodate the separation of the standard MedBiquitous abstract (interface) layer WSDL files and associated member-specific binding implementation files, accommodating a single point of entry and simple central registry for service discovery. (Note: This implies space and id [FTP ?, etc] management by the MedBiquitous staff for member organizations. An alternate approach is hosting binding details on each member organization's site with the disadvantage of a single point of entry for service discovery).

Concerning service discovery, MedBiquitous will initially rely on manual development time (static) discovery by developers into the single point of entry of the medbiq.org service taxonomy. There will be no runtime dynamic binding API dependency on a registry mechanism. When a member organization wants to call a Web service provided by another, the developer searches the taxonomy provided on medbiq.org for an implementation by the member with which he needs to interact, and downloads the implementation WSDL, which drags along the abstract interface WSDL MedBiquitous standard. The developer creates a client that calls the Web service.

10.7 Binary Information Transmit

MedBiquitous will leverage the SOAP Message Transmission Optimization Mechanism [[MTOM](#)] specification for transmitting binary information. Implementations leveraging MTOM should use WSI Basic Profile 1.2 [[WSIBP1.2](#)].

11. References

[keywords]

S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, Harvard University, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>

[URI]

T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," RFC 3986, W3C/MIT, Day Software, Adobe Systems, January 2005. <http://www.ietf.org/rfc/rfc3986.txt>

[XPath]

XML Linking Language (XLink) Version 1.0, <http://www.w3.org/TR/2001/REC-xlink-20010627/>

[XSLT]

XSL Transformations (XSLT) Version 1.0, <http://www.w3.org/TR/xslt>

MedBiquitous	Version: 2.0
Web Services Design Guidelines	Date: 13 May 2009

[XMLNamespace]

W3C Recommendation, Namespaces in XML, <http://www.w3.org/TR/1999/REC-xml-names-19990114>

[XML]

Extensible Markup Language (XML) 1.0 (Second Edition), <http://www.w3.org/TR/REC-xml>

[XMLSchema1]

W3C Recommendation, XML Schema Part 1: Structures, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

[XMLSchema2]

W3C Recommendation, XML Schema Part 2: Datatypes, <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

[WSIBP]

Web Services Interoperability Basic Profile 1.1, <http://www.ws-i.org/Profiles/BasicProfile-1.1.html>

[RFC2045]

A Multipurpose Internet Mail Extensions (MIME) Part One, Format of Internet Message Bodies, <http://www.ietf.org/rfc/rfc2045.txt?number=2045>

[RFC2046]

A Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, <http://www.ietf.org/rfc/rfc2046.txt?number=2046>

[WS-I Basic Security]

Web Services Interoperability Basic Security Profile 1.0. <http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html>

[MTOM]

Message Transmission Optimization Mechanism, <http://www.w3.org/TR/soap12-mtom/>

[WS-Security]

Web Services Security. <http://www.oasis-open.org/specs/#wssv1.1>

[MedBiquitous]

MedBiquitous, Collaborative technologies for medical education, <http://www.medbiq.org/>

[WSDL]

W3C Note, Web services Description Language 1.1, <http://www.w3.org/TR/wsdl>

A1.

[WSIBP1.2]

Web Services Interoperability Basic Profile 1.2, <http://www.ws-i.org/Profiles/BasicProfile-1.2.html>