

NTCIP 2306 v01.51

National Transportation Communications for ITS Protocol Application Profile for XML Message Encoding and Transport in ITS Center to Center Communications (NTCIP C2C XML)

v00.50 March 1, 2005
v01.51 March 29, 2005

This is a draft document, which is distributed for review and comment purposes only. You may reproduce and distribute this document within your organization, but only for the purposes of and only to the extent necessary to facilitate review and comment to the **NTCIP Coordinator**. Please ensure that all copies include this notice. This document contains preliminary information that is subject to change.

Published by

American Association of State Highway and Transportation Officials (AASHTO) [10 pt and bold]
444 North Capitol Street, N.W., Suite 249
Washington, D.C. 20001

Institute of Transportation Engineers (ITE)
1099 14th Street, N.W., Suite 300 West
Washington, D.C. 20005-3438

National Electrical Manufacturers Association (NEMA)
1300 North 17th Street, Suite 1847
Rosslyn, Virginia 22209-3801

Draft Document Revision History

Title	Description	Version	Date
NTCIP C2C XML WG Draft Apr 27 2004	Initial WG Draft Document Creation by Consultants	00.10	April 27, 2004
NTCIP C2C XML WG Draft May 6 2004	Initial WG Draft for Distribution to WG	00.20	May 6, 2004
NTCIP C2C XML WG Draft May 27, 2004	NTCIP 2306 number assigned at NTCIP JC Meeting	00.21	May 27, 2004
NTCIP C2C XML WG Draft v2 Oct 27 2004	Document Re-organization	00.30	Oct 27, 2004
NTCIP C2C XML WG Draft v2 Nov 16 2004	Reformatted per NTCIP standard format. Response to working group comments.	00.40	Nov 16, 2004
NTCIP C2C XML WG Draft v2 Dec 10 2004	Disposition of Comments. Reorganization of document into sub profiles. Added PRL.	00.41	Dec 10, 2004
NTCIP C2C XML WG Draft v2 Feb 14 2005	Second WG Draft for Distribution to WG	00.42	Feb 14, 2005
NTCIP C2C XML UCD Draft v1 Mar 1 2005	Incorporated WG comments on 2 nd WG draft & first User Comment Draft for distribution to JC.	00.50	Mar 1, 2005
AP for XML Message Encoding and Transport in ITS C2C Communications	Schopp added and reformatted title page and front matter for UCD distribution after acceptance by the JC 3/17/05.	01.51	3/29/05

NOTICES

Copyright Notice

© 2005 by the American Association of State Highway and Transportation Officials (AASHTO), the Institute of Transportation Engineers (ITE), and the National Electrical Manufacturers Association (NEMA). All intellectual property rights, including, but not limited to, the rights of reproduction, translation and display are reserved under the laws of the United States of America, the Universal Copyright Convention, the Berne Convention, and the International and Pan American Copyright Conventions. Except as licensed or permitted, you may not copy these materials without prior written permission from AASHTO, ITE, or NEMA. Use of these materials does not give you any rights of ownership or claim of copyright in or to these materials.

Visit www.ntcip.org for other copyright information, for instructions to request reprints of excerpts, and to request reproduction that is not granted below.

[NTCIP Coordinator NOTE: For each excerpt from other works, permission will be requested and obtained before publication.] This standard uses examples drawn from the XML schemas and the message sets developed by other standards organizations engaged in the FHWA cooperative development. The use of portions of these messages in examples provided in this standard does not grant any rights of ownership or claim of copyright in or to these materials.

PDF File License Agreement

To the extent that these materials are distributed by AASHTO / ITE / NEMA in the form of an Adobe® Portable Document Format (PDF) electronic data file (the "PDF File"), AASHTO / ITE / NEMA authorizes each registered PDF File user to view, download, copy, or print the PDF File available from the authorized Web site, subject to the terms and conditions of this license agreement:

- (a) you may download one copy of each PDF File for personal, noncommercial, and intraorganizational use only;
- (b) ownership of the PDF File is not transferred to you; you are licensed to use the PDF File;
- (c) you may make one more electronic copy of the PDF File, such as to a second hard drive or burn to a CD;
- (d) you agree not to copy, distribute, or transfer the PDF File from that media to any other electronic media or device;
- (e) you may print one paper copy of the PDF File;
- (f) you may make one paper reproduction of the printed copy;
- (g) any permitted copies of the PDF File must retain the copyright notice, and any other proprietary notices contained in the file;
- (h) the PDF File license does not include (1) resale of the PDF File or copies, (2) republishing the content in compendiums or anthologies, (3) publishing excerpts in commercial publications or works for hire, (4) editing or modification of the PDF File except those portions as permitted, (5) posting on network servers or distribution by electronic mail or from electronic storage devices, and (6) translation to other languages or conversion to other electronic formats;
- (i) other use of the PDF File and printed copy requires express, prior written consent.

Content and Liability Disclaimer

The information in this publication was considered technically sound by the consensus of persons engaged in the development and approval of the document at the time it was developed. Consensus does not necessarily mean that there is unanimous agreement among every person participating in the development of this document.

AASHTO, ITE, and NEMA standards and guideline publications, of which the document contained herein is one, are developed through a voluntary consensus standards development process. This process brings together volunteers and/or seeks out the views of persons who have an interest in the topic covered by this publication. While AASHTO, ITE, and NEMA administer the process and establish rules to promote fairness in the development of consensus, they do not write the document and they do not independently test, evaluate, or verify the accuracy or completeness of any information or the soundness of any judgments contained in their standards and guideline publications.

AASHTO, ITE, and NEMA disclaim liability for any personal injury, property, or other damages of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, application, or reliance on this document. AASHTO, ITE, and NEMA disclaim and make no guaranty or warranty, express or implied, as to the accuracy or completeness of any information published herein, and disclaims and makes no warranty that the information in this document will fulfill any of your particular purposes or needs. AASHTO, ITE, and NEMA do not undertake to guarantee the performance of any individual manufacturer or seller's products or services by virtue of this standard or guide.

In publishing and making this document available, AASHTO, ITE, and NEMA are not undertaking to render professional or other services for or on behalf of any person or entity, nor are AASHTO, ITE, and NEMA undertaking to perform any duty owed by any person or entity to someone else. Anyone using this document should rely on his or her own independent judgment or, as appropriate, seek the advice of a competent professional in determining the exercise of reasonable care in any given circumstances. Information and other standards on the topic covered by this publication may be available from other sources, which the user may wish to consult for additional views or information not covered by this publication.

AASHTO, ITE, and NEMA have no power, nor do they undertake to police or enforce compliance with the contents of this document. AASHTO, ITE, and NEMA do not certify, test, or inspect products, designs, or installations for safety or health purposes. Any certification or other statement of compliance with any health or safety-related information in this document shall not be attributable to AASHTO, ITE, or NEMA and is solely the responsibility of the certifier or maker of the statement.

Trademark Notice

NTCIP is a trademark of AASHTO / ITE / NEMA. All other marks mentioned in this document are the trademarks of their respective owners.

ACKNOWLEDGEMENTS

This publication was prepared by the NTCIP Center-to-Center Working Group, which is a subdivision of the Joint Committee on the NTCIP. The Joint Committee is organized under a Memorandum of Understanding among the American Association of State Highway and Transportation Officials (AASHTO), the Institute of Transportation Engineers (ITE), and the National Electrical Manufacturers Association (NEMA). The Joint Committee on the NTCIP consists of six representatives from each of the standards organizations, and provides guidance for NTCIP development.

At the time that this document was prepared, the following individuals were members of the NTCIP Center-to-Center Working Group:

- Michael Brown
- Jeff Brummond
- Manny Insignares (Chair)
- Timothy A. Johnson
- David Kelley
- Scott Lee
- Jeff Mayo
- James Ort
- Brian Pecus
- Jeffrey Thorn
- Warren Tighe
- Ken Vaughn

Other individuals providing input to the document include:

- Blake Christie

In addition to the many volunteer efforts, recognition is also given to those organizations that supported the efforts of the working groups by providing comments and funding for the standard, including:

- ConSysTec
- Siemens ITS
- Southwest Research Institute
- SubCarrier Systems
- Caltrans
- General Dynamics
- Illinois DOT
- Iteris, Inc.
- Minnesota DOT
- Mitretek
- New York State DOT
- TransCore
- Trevilon Corp.

Recognition is also given to the liaisons from the ITS message set standards who have supported the efforts of the NTCIP working group so that the resulting standard may be used with the message set standards they have produced. This includes working group committee members of the following: Advanced Traffic Management Data Dictionary and External Message Sets for Traffic Management Center Communications (an ITE/AASHTO joint standard), the Transit Communications Interface Profile (a standard of APTA), the Message Sets for Advanced Traveler Information Systems (SAE-J2354) (a standard of the Society of Automotive Engineers), IEEE-1512 - Standard for Common Incident Management Message Sets for Use by Emergency Management Centers (a standard of IEEE).

FOREWORD

This document uses only metric units.

This document defines an application profile for communications between transportation management systems, using Internet standards based on the Extensible Markup Language (XML). It defines requirements and optional and conditional clauses that are applicable to specific environments for which they are intended.

This document is an NTCIP Application Profile. An NTCIP Application Profile defines the upper three layers of the ISO seven-layer Open Systems Interconnect (OSI) Reference Model. A Joint NTCIP Application Profile standards publication is equivalent to these document types at the standards organizations:

AASHTO – [Standard Specification]
ITE – [Software Standard]
NEMA – [Standard]

For more information about NTCIP standards, visit the NTCIP Web Site at <http://www.ntcip.org>.

User Comment Instructions

The term “User Comment” includes any type of written inquiry, comment, question, or proposed revision, from an individual person or organization, about any part of this standard publication’s content. A “Request for Interpretation” of this standard publication is also classified as a User Comment. User Comments are solicited at any time. In preparation of this NTCIP standards publication, input of users and other interested parties was sought and evaluated.

All User Comments will be referred to the committee responsible for developing and/or maintaining this standards publication. The committee chairperson, or their designee, may contact the submitter for clarification of the User Comment. When the committee chairperson or designee reports the committee’s consensus opinion related to the User Comment, that opinion will be forwarded to the submitter. The committee chairperson may report that action on the User Comment may be deferred to a future committee meeting and/or a future revision of the standards publication. Previous User Comments and their disposition may be available for reference and information at www.ntcip.org.

A User Comment should be submitted to this address:

NTCIP Coordinator
National Electrical Manufacturers Association
1300 North 17th Street, Suite 1847
Rosslyn, Virginia 22209-3801
fax: (703) 841-3331
e-mail: ntcip@nema.org

A User Comment should be submitted in the following form:

Standard Publication number and version:
Page:
Paragraph or Clause:
Comment:

Please include your name, organization, and address in your correspondence.

Approvals

This standards publication was separately balloted and approved by AASHTO, ITE, and NEMA after recommendation by the Joint Committee on the NTCIP. Each organization has approved this standard as the following standard type, as of the date:

AASHTO – [Standard Specification; Month 200X]
ITE – [Software Standard; Month 200X]
NEMA – [Standard; Month 200X]

History

This standard has not been previously officially released in any form or under any other name.

Compatibility of Versions

All NTCIP Standards Publications have a major and minor version number for configuration management. The version number syntax is "v00.00a," with the major version number before the period, and the minor version number and edition letter (if any) after the period.

Anyone using this document should seek information about the version number that is of interest to them in any given circumstance. The MIB, the PRL, and the PICS should all reference the version number of the standards publication that was the source of the excerpted material.

Compliant systems based on later, or higher, version numbers MAY NOT be compatible with compliant systems based on earlier, or lower, version numbers. Anyone using this document should also consult NTCIP 8004 for specific guidelines on compatibility.

INTRODUCTION

This is a **User Comment Draft** of NTCIP 2306, ITS Center-to-Center Communications Profile for XML Message Encoding and Transport (hereafter referred to as NTCIP C2C XML). This document represents the first of two identified phases for the development of an XML communications standard for center-to-center communications in the transportation domain (real-time management of public roads and transit systems).

The purpose of this standard is to allow transportation agencies and center managers to specify and implement communications interfaces (message form, message usage, and transport) for transmitting information encoded in the Extensible Markup Language (XML) between their center and an external center. Message content is defined in other standards, such as the Advanced Traffic Management Data Dictionary and External Message Sets for Traffic Management Center Communications (an ITE/AASHTO joint standard), the Transit Communications Interface Profile (a standard of APTA), the Message Sets for Advanced Traveler Information Systems (SAE-J2354) (a standard of the Society of Automotive Engineers), IEEE-1512 - Standard for Common Incident Management Message Sets for Use by Emergency Management Centers (a standard of IEEE), and other standards defining XML messages content in the transportation domain.

A number of projects have begun to design and/or implement XML as part of their center-to-center communications efforts. This interim draft standard is being developed to provide projects a reasonably complete standard with which to specify XML message usage and transport, and to provide a basis for testing of center system interfaces that communicate with external centers.

This standard defines requirements that are applicable to all environments using NTCIP XML for center-to-center communications, and it also contains optional and conditional clauses that are applicable only to specific environments for which they are intended.

The following keywords apply to this document: AASHTO, ITE, NEMA, NTCIP, XML, WSDL, SOAP, C2C, data, message, center-to-center.

In 1992, the NEMA 3-TS Transportation Management Systems and Associated Control Devices Section began the effort to develop the NTCIP. The Transportation Section's purpose was to respond to user needs to include standardized systems communication in the NEMA TS 2 standard, *Traffic Controller Assemblies*. Under the guidance of the Federal Highway Administration's NTCIP Steering Group, the NEMA effort was expanded to include the development of communications standards for all transportation field devices that could be used in an Intelligent Transportation Systems (ITS) network, and to center-to-center communications.

In 1996, an agreement was executed among AASHTO, ITE, and NEMA to jointly develop, approve, and maintain the NTCIP standards. In July, 1996, the NTCIP Center-to-Center Working Group met for the first time, and began the effort to standardize C2C protocols. The working group first defined the DATEX-ASN and CORBA standards for center-to-center communications. In 2003, work began on a third center-to-center protocol based on web services and the Extensible Markup Language (XML).

CONTENTS

Section 1 General	1
1.1 Scope.....	1
1.2 References	1
1.2.1 Normative References.....	1
1.2.2 Informative References	2
1.3 Document Organization.....	2
1.4 Conformance Clause.....	3
1.5 Glossary of Terms	3
Section 2 C2C Messaging Needs and Derived Requirements	5
2.1 C2C Communications Environments.....	5
2.2 XML Profile Needs.....	5
2.3 XML Profile Requirements.....	6
Web Services Sub Profile Descriptions	7
Section 3 NTCIP C2C XML Profile Solutions	10
3.1 Sub Profile Solution Descriptions	10
3.2 Traceability to C2C XML Profile Requirements.....	10
Section 4 Message Encoding	13
4.1 XML Message Encoding.....	13
4.1.1 XML Text Encoding	13
4.1.2 XML Gzip Encoding.....	13
4.2 SOAP Message Encoding	13
Section 5 Message Transport	15
5.1 Message Transport Using HTTP	15
5.1.1 HTTP Headers for XML Text and Gzip	15
5.1.2 HTTP Headers for SOAP	15
5.1.3 HTTP Secure Sockets (HTTPS).....	15
5.1.4 HTTP Version.....	15
5.2 Message Transport Using FTP.....	15
5.2.1 FTP.....	15
Section 6 Sub Profiles Common	16
6.1 WSDL Common.....	16
6.1.1 General Requirements	16
6.1.2 Required WSDL Sections.....	16
6.1.3 Profile Common WSDL Sections	16
6.1.4 Sub Profile Specific WSDL Sections.....	17
6.2 WSDL Definitions Section	18
6.3 WSDL Types and Schema Section	18
6.4 WSDL Message Sections.....	19

6.5	Description of the Publication-Subscription MESSAGE Transmission Patterns	20
6.5.1	Subscription	20
6.5.2	Publication	20
6.5.3	Responsibilities of the Message Set SDOs	21
6.6	Description of Transport for Secure Socket Service Endpoints	21
6.7	Specifying Additional Web Service Items	21
Section 7 WSDL for Sub Profile SOAP Over HTTP		22
7.1	WSDL for SOAP Request / Response	22
7.1.1	WSDL for SOAP Request-Response PortType Section	22
7.1.2	WSDL for SOAP Request-Response Binding Section	22
7.1.3	WSDL for SOAP Request-Response Service Section	23
7.2	WSDL for SOAP Subscriber's Callback Listener (Publish-Subscribe)	24
7.2.1	WSDL for SOAP Subscriber's Callback Listener PortType Section	24
7.2.2	WSDL for SOAP Subscriber's Callback Listener Binding Section	24
7.2.3	WSDL for SOAP Subscriber's Callback Listener Service Section	25
Section 8 WSDL for Sub Profile XML Over HTTP		26
8.1	WSDL for XML over HTTP (Request Only File Retrieval)	26
8.1.1	WSDL for XML Over HTTP Request Only PortType Section	26
8.1.2	WSDL for XML Over HTTP Request Only Binding Section	26
8.2	WSDL for XML over HTTP (Request-Response)	27
8.2.1	WSDL for XML Over HTTP Request-Response PortType Section	27
8.2.2	WSDL for XML Over HTTP Request-Response Binding Section	27
8.3	WSDL for XML over HTTP Service Section	28
Section 9 WSDL for Sub Profile XML Over FTP		29
9.1	WSDL for XML over FTP (FILE RETIEVAL)	29
9.1.1	Extending WSDL for FTP	29
9.1.2	WSDL for XML over FTP PortType Section	29
9.1.3	WSDL for XML over FTP Binding Section	29
9.1.4	WSDL for XML over FTP Service Section	30
Section 10 Test Plan Development and Conformance Guidance		31
10.1	WSDL Development Methodology	31
10.2	Testing the Correctness of the WSDL	31

SECTION 1 GENERAL

1.1 SCOPE

This standard is applicable to communications between any two management subsystems, also called centers, within the Intelligent Transportation Systems (ITS) environment.

This application profile lists the requirements for use of XML and related protocols for data exchange among ITS management systems.

1.2 REFERENCES

For approved amendments, contact the NTCIP Coordinator at the address shown above.

For draft amendments of this document, which are under discussion by the relevant NTCIP Working Group, and recommended amendments of the NTCIP Joint Committee, visit the World Wide Web at <http://www.ntcip.org>.

Normative references contain provisions which, through reference in this text, constitute provisions of this Standard. Other documents and standards (other references) referenced in these documents provide background information that may help the reader gain a complete understanding of the profile and the relations between parts of the profile. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

1.2.1 Normative References

Table 1-1. Normative References

Organization	Document Title	Date
WS-I	Basic Profile Version 1.0a – Final Specification	August 8, 2003
W3C	Web Services Description Language (WSDL) 1.1 – W3C Note	March 15, 2001
W3C	Extensible Markup Language (XML) 1.0	February 4, 2004
W3C	Simple Object Access Protocol (SOAP) 1.1	May 8, 2000
IETF	Hypertext Transfer Protocol HTTP/1.1 RFC 2616	June 1999
IETF	File Transfer Protocol (FTP) RFC 959	October 1985

The WS-I Basic Profile Requirements are constrained to meet the C2C XML Needs, or removed where no clear need has been identified. Appendix C identifies the requirements (and constraints applied) of the WS-I Basic Profile.

In the event of conflicts between multiple standards referenced here, the order of document precedence shall be as follows:

1. Application Profile for XML in C2C (this standard)
2. WS-I Basic Profile
3. W3C WSDL
4. SOAP 1.1
5. XML
6. HTTP & FTP

1.2.2 Informative References

Table 1-2. Informative References

Organization	Document Title	Date
NTCIP	NTCIP 9010 v01.09 – Information Report, XML in Center-to-Center Communications	November 23, 2004
ITE/AASHTO	Standards for Traffic Management Center to Center Communications Volume II, Message Tables & Sequence Diagrams	December 15, 2003
WS-I	WS-I Usage Scenarios Version 1.01 – Final Specification	December 9, 2003

1.3 DOCUMENT ORGANIZATION

This document is comprised of ten Sections and three Appendices.

- Section 1 - General. This section provides an overview of the document and introductory material.
- Section 2 - C2C Messaging Needs and Derived Requirements. This section identifies this Profile's needs as defined by the TMDD Volume II (Dialogs), and the NTCIP 9010 Information Report – XML in ITS Center-to-Center Communications.
- Section 3 - NTCIP C2C XML Profile Solutions. This section contains 1) a trace of requirements to solutions, and 2) a trace of solutions (sub profiles) to the document sections that apply (PRL).
- Section 4 - Message Encoding. This section provides a normative reference for the encoding of XML and SOAP messages.
- Section 5 - Message Transport. This section provides a normative reference for transport of XML and SOAP encoded messages over the HTTP and FTP.
- Section 6 - Sub Profiles Common. This section provides a normative reference for the description of a web service using the WSDL, message encoding, and transport common across the sub profiles,
- Section 7 - WSDL Sub Profile for SOAP over HTTP. This section provides a normative reference for the description of a web service using the WSDL for SOAP encoded messages using the HTTP. This section supports 2 transmission patterns: 1) request-response, and 2) publish-subscribe.
- Section 8 - WSDL Sub Profile for XML over HTTP. This section provides a normative reference for the description of a web service using the WSDL for XML encoded messages using the HTTP. This section supports the transmission of XML over HTTP with 2 transmission patterns: 1) request only, and 2) request-response.
- Section 9 - WSDL Sub Profile for XML over FTP. This section provides a normative reference for the description of a web service using the WSDL for file transfer of XML encoded messages using the FTP. This section supports 1 transmission patterns: 1) request only.
- Section 10 - Test Plan Development and Conformance Guidance. This section provides a brief overview of 1) WSDL development, 2) how WSDL can be used in the development of more comprehensive test plans for testing of center interfaces that use this standard, and therefore the WSDL, and 3) methods for testing the correctness of the syntax of the WSDL that conforms with this standard.
- Appendix A – Sample WSDL Worksheet for TMDD Dialogs. Provides examples of TMDD dialogs and the XML version of the TMDD messages involved in each.
- Appendix B - Sample WSDL for TMDD. Contains an example WSDL document based on the contents of Appendix A.
- Appendix C - WS-I Basic Profile Requirements Constraints. Identifies constraints applicable to each of the requirements specified in the WS-I Basic Profile.

1.4 CONFORMANCE CLAUSE

A center system conforms with this specification as follows:

1. WSDL document(s) exist that conform with the normative sections of this standard, and XML Schema documents referenced by the WSDL exist.
2. The WSDL syntax can be shown to be correct using one of the methods outlined in Section 10.
3. A center system can be shown to support the web services described in the conforming WSDL, including: 1) XML messages validate against a referenced schema, 2) message inputs and outputs described in the WSDL can be reproduced under a system test, 3) messages can be exchanged using the transports defined in the WSDL.

1.5 GLOSSARY OF TERMS

The following glossary defines terms and acronyms used in this standard.

Table 1-3. Glossary of Terms

Term	Definition
AASHTO	American Association of State Highway and Transportation Officials
ANSI	American National Standards Institute
AP	Application Profile
APTA	American Public Transportation Association
ASP	Active Server Pages
ATIS	Advanced Traveler Information Systems
C2C	Center-to-Center
DSRC	Dedicate Short Range Communications
FHWA	Federal Highway Administration
FTP	File Transfer Protocol
GNU	GNU is Not Unix
GZIP	GNU Zip -- a compression utility
HTTP	Hypertext Transfer Protocol
HTTPS	Secure HTTP (Hypertext Transfer Protocol)
IEEE	Institute of Electrical and Electronics Engineers
IEEE 1512	An IEEE family of message set standards for Incident Management
IETF	Internet Engineering Task Force
IM	Incident Management
ITE	Institute of Transportation Engineers
ITIS	International Traveler Information System (SAE J2540-1)
ITS	Intelligent Transportation Systems
JSP	Java Server Pages
LRMS	Location Referencing Message Specification
MIME	Multipurpose Internet Mail Extensions (IETF RFC 1521)
MS/ETMCC	Message Sets for External Traffic Management Center Communications
NEMA	National Electrical Manufacturers Association
NTCIP	National Transportation Communications for ITS Protocol
NTCIP 9010	NTCIP Information Report on ITS Center-to-Center Communications
OASIS	Organization for the Advancement of Structured Information Standards
PERL	Practical Extraction and Report Language
PHP	Hypertext Preprocessor
PN	Profile Need
PR	Profile Requirement
RFC	Request for Comment

Term	Definition
SAE	Society of Automotive Engineers
SAE J2354	An SAE Message Set standard for Advanced Traveler Information
SDO	Standards Development Organization
SOAP	Simple Object Access Protocol
TCIP	Transit Communications Interface Profile, a standard of APTA
TMDD	Traffic Management Data Dictionary, a joint standard of ITE/AASHTO
UDDI	Universal Description, Discovery and Integration
W3C	World Wide Web Consortium
WS-I	Web Services Interoperability Organization
WSDL	Web Services Description Language
XML	eXtensible Markup Language

SECTION 2 C2C MESSAGING NEEDS AND DERIVED REQUIREMENTS

2.1 C2C COMMUNICATIONS ENVIRONMENTS

This document uses examples based on the concept of operations, dialogs, and messages defined in ITE/AASHTO *Standards for Traffic Management Center to Center Communications* (TMDD). Similar messaging patterns can be found in the message set efforts of the other standards. The messages defined in the TMDD effort are used here as representative examples in the text that follows.

The dialogs and messaging patterns represented in the TMDD vary in complexity.

The following is an example of a simple C2C environment:

A center just wants to periodically retrieve one type of information (e.g., current incident data) from another center.

The following is an example of a complex C2C environment:

A center wants to be automatically notified of new incidents as they occur, to retrieve traffic volume and traffic signal status data for the roadways adjacent to the incident, and sometimes to request the other center to change its signal timing plan or post a particular message on a nearby changeable message sign.

The NTCIP 9010 Information Report – XML in ITS Center-to-Center Communications, identified two different XML-based mechanisms, one suited to simple C2C environments, and the other suited to complex environments.

The complex mechanism is called XML SOAP, and involves the use of SOAP (Simple Object Access Protocol) and WSDL (Web Services Description Language). This solution supports subscription services, where a center can make a one-time request to another center that results in a standard message being sent automatically whenever data changes. It also supports commands, or action requests, being sent to another center. In this example, messages sent in response to a subscription are supplier initiated (the center providing the data initiates the exchange), while commands sent in the other direction are consumer initiated. The various data frames found in message sets to support different styles and content for subscriptions fits well with this paradigm.

The simple mechanism is called XML Direct, and allows a center to offer information to other centers by simply making it available as an XML file (document) at a known web address. The file contains one or more TMDD messages encoded as XML. The center providing the information updates this file when any data change. Other centers retrieve the file using plain HTTP or FTP whenever they need such information, or retrieve it regularly in order to monitor it for changes. All such exchanges are always data consumer-initiated. Using XML Direct, a center (data provider) can make information available, and other centers can retrieve it, without either party having to implement the more complex software needed to support XML SOAP. This represents a significant cost saving for centers that do not need to support complex messaging. And it is suitable for many legacy practices found in deployments today.

2.2 XML PROFILE NEEDS

Profile needs relevant to simple C2C environments (those satisfied by the XML Direct mechanism) are relatively straight forward. This mechanism needs only support consumer-initiated file transfers. Data publishing practices (file update rates, groupings of content, etc.) are typically known between the involved systems. However, because of its simplicity, no methods for the support for data recovery in the event of loss by the data consumer are provided.

Needs relevant to XML SOAP were derived from the ITE/AASHTO *Standards for Traffic Management Center to Center Communications Volume II, Message Tables & Sequence Diagrams* (referred to hereafter as TMDD dialogs), and through liaison activities with working group and committee members of other message set standards.

A review of the TMDD messages and dialogs (as described by the sequence diagrams and accompanying written material) indicate the messaging patterns and encoding needs described in the table below. Sample TMDD messages and dialogs are further described in Appendix A. The needs of other center-to-center domains and message sets (e.g., ATIS, Incident Management, Transit) have been found to be similar, and this standard is intended to be applicable within and between all ITS domains.

The need for secure connections derives from the use of the Internet for center-to-center communications that may involve transfers of sensitive data and device commands.

Table 2-1. Profile Needs

Profile Need (PN)	Description	Need is Covered in this Standard
PN 1.0	Message and File Transfer	
PN 1.1	Get File. Allow the information consumer center to initiate transfer of a file from the information supplier center.	Y
PN 2.0	Message Encoding	
PN 2.1	ASN.1 or XML. For this profile, messages need to be encoded as XML.	Y
PN 2.2	Privacy. Allow messages to be privately transmitted over the Internet and other shared networks.	Y
PN 3.0	Message Exchange Patterns	
PN 3.1	Single Request-Response. This message transmission pattern describes communication of a single message sent from one center (the request) and a single message response from the other center.	Y
PN 3.2	Subscription. This message transmission pattern describes the ability of a center to subscribe for automatic and on-going publication of information from another center. A subscription is established using either a Single Request-Response or Multiple Request-Response message transmission pattern. The subsequent publication messages are the subject of the following need.	Y
PN 3.3	Publication. This message transmission pattern describes the communication of an asynchronous response message, from a center that provides or publishes information, to another center that has previously subscribed for that information. This exchange is initiated by the publisher.	Y

2.3 XML PROFILE REQUIREMENTS

The following table describes the XML profile requirements that derive from the above needs.

Table 2-2. Profile Requirements

Profile Requirement (PR)		PN	Requirement Dependencies
PR 1.0	Message Encoding		
PR 1.1	XML Encoding. The profile shall exchange messages that are encoded as XML.	PN 2.1	
	a) Text Encoding	PN 2.1	
	b) Gzip MIME Encoding.	PN 2.1	
PR 1.2	SOAP Encoding	PN 2.1	
PR 2.0	Message Transport		
PR 2.1	HTTP. The profile shall provide a mechanism to describe and accomplish the exchange of XML using the HTTP.	PN 1.1, 3.1, 3.2, 3.3	
	a) HTTP	PN 1.1, 3.1, 3.2, 3.3	
	b) HTTPS	PN 1.1, 3.1, 3.2, 3.3, 2.2	
PR 2.2	FTP. The profile shall provide a mechanism to describe and accomplish the exchange of XML documents using the FTP.	PN 1.1	
PR 3.0	Web Services General		
PR 3.1	Referencing External XML Schema. The profile shall provide a mechanism to reference one or more external schemas, which define a message set for deployment.	PN 1.1, 2.1, 3.1, 3.2, 3.3	
PR 3.2	a) XML Schema Version and Date. The profile shall provide a mechanism to describe the version and date of the referenced schema.	PN 1.1, 2.1, 3.1, 3.2, 3.3	PR 3.1
PR 3.3	Define which XML Schema Messages are supported. The profile shall provide a mechanism to describe which specific messages of the message set are supported by a center.	PN 1.1, 3.1, 3.2, 3.3	
PR 3.4	Frequency of XML file updates. The profile shall provide a mechanism to describe the frequency of XML file updates.	PN 1.1	
PR 4.0	Web Services Sub Profile Descriptions		

Profile Requirement (PR)		PN	Requirement Dependencies
PR 4.1	Request-Response Message Transmission Pattern. The profile shall provide a mechanism to describe and accomplish the transmission between centers of a request message followed by a response message.	PN 3.1	
PR 4.1.1	Define Interfaces – Operations and Message Inputs and Outputs. The profile shall provide a mechanism to describe the operations and associated message inputs and outputs for a request-response message pair.	PN 3.1	PR 3.3
PR 4.1.2	Define Message Encoding and Transport. The profile shall provide a mechanism to describe and accomplish a request-response message transmission pattern for a) SOAP XML messages transmitted over the HTTP protocol, and b) XML messages transmitted over HTTP.	PN 2.1, 3.1	
	a) SOAP/HTTP	PN 2.1, 3.1	PR 1.1 PR 2.1
	b) XML/HTTP	PN 2.1, 3.1	PR 1.2 PR 2.1
PR 4.2	Subscription Message Transmission Pattern. The profile shall provide a mechanism to describe and accomplish one center to subscribe to a periodic or event-driven publication of another center. The subscription pattern shall be implemented as a request-response pattern.	PN 3.2	
PR 4.2.1	Define Interfaces – Operations, Message Inputs and Outputs. The profile shall provide a mechanism to describe the operation(s) and message inputs and outputs used in a subscription message transmission pattern.	PN 3.2	PR 3.3
PR 4.2.2	Defining Message Encoding and Transport. The profile shall provide a mechanism to describe and accomplish a subscription transmission for a) SOAP encoded messages transmitted over the HTTP protocol, and b) XML messages transmitted over HTTP.	PN 2.1, 3.2	
	a) SOAP/HTTP	PN 2.1, 3.2	PR 1.2

Profile Requirement (PR)		PN	Requirement Dependencies
			PR 2.1
	b) XML/HTTP	PN 2.1, 3.2	PR 1.1 PR 2.1
PR 4.3	Publication Message Transmission Pattern. The profile shall provide a mechanism for a center to transmit asynchronous responses to a subscription.	PN 3.3	
PR 4.3.1	Define Interfaces – Operations, Message Inputs and Outputs. The profile shall provide a mechanism to describe the operation(s) and message inputs and outputs used by a subscription to receive a publication transmission.	PN 3.3	PR 3.3
PR 4.3.2	Defining Message Encoding and Transport. The profile shall provide a mechanism to describe and accomplish the publication transmission for a) SOAP XML messages transmitted over the HTTP protocol, and b) XML messages transmitted over HTTP.	PN 2.1, 3.2	
	a) SOAP/HTTP	PN 2.1, 3.2	PR 1.1 PR 2.1
	b) XML/HTTP	PN 2.1, 3.2	PR 1.2 PR 2.1
PR 4.4	File Retrieval (Request Only Pattern, XML Direct). The profile shall provide a mechanism to define and a file for retrieval by another center.	PN 1.1	
PR 4.4.1	Define Interfaces – Message Outputs. The profile shall provide a mechanism to describe the operation(s) and message outputs used to retrieve a file by another center.	PN 1.1	PR 3.3
PR 4.4.2	Define Message Encoding and Transport. The profile shall provide a mechanism to describe and accomplish the retrieval of a) an XML file using the FTP, and b) an XML message using the HTTP.	PN 1.1, 2.1	
	a) XML using HTTP 'Get'	PN 1.1, 2.1	PR 1.1 PR 2.1
	b) XML using FTP 'Get'	PN 1.1, 2.1	PR 1.1 PR 2.2

SECTION 3 NTCIP C2C XML PROFILE SOLUTIONS

3.1 SUB PROFILE SOLUTION DESCRIPTIONS

The AP-C2CXML is divided into 3 separate sub profiles. A sub profile is an implementation bundle that consists of the following information:

- 1) WSDL Definition Requirements (the format of the WSDL, not the WSDL itself)
- 2) Message Encoding
- 3) Message Transport

The sub profiles of the AP-C2C XML are:

- 1.0 SOAP over HTTP
- 2.0 XML over HTTP
- 3.0 XML over FTP

3.2 TRACEABILITY TO C2C XML PROFILE REQUIREMENTS

The following table shows the solutions that the profile shall use to meet each of the profile requirements. The table represents the NTCIP 2306 Profile Requirements List (PRL). The column labeled 'NTCIP 2306 Section' contains additional normative material found in this document. The table facilitates the ability to customize the requirements and normative materials in this standard based on a sub profile that matches the design criteria and requirements of a specific implementation. A specific implementation is expected to select one or more of the sub profiles.

A 'Project Requirement' column has been added to facilitate use of the PRL by specification and PICS developers.

Table 3-1. Profile Requirements to Solution Trace (Profile Requirements List)

	Profile Requirements List (PRL)	NTCIP 2306 Section	Mandatory / Optional	Profile Requirement	Project Requirement
1.0	SOAP over HTTP				
	a) WSDL Request-Response		M		
	- WSDL General	6.1	M		
	- Definitions	6.2	M	PR 3.1	
	- Types/Schema	6.3	M	PR 3.1, 3.2	
	- Message	6.4	M	PR 3.3	
	- PortType (Interfaces)	7.1.1	M	PR 4.1.1	
	- Binding (Transport)	7.1.2	M	PR 4.1.2a	
	- Service (Transport)	7.1.3	M	PR 4.1.2a	
	b) WSDL Publish-Subscribe		O		
	- WSDL General	6.1	M		
	- Definitions	6.2	M	PR 3.1	
	- Types/Schema	6.3	M	PR 3.1, 3.2	
	- Message	6.4	M	PR 3.3	
	- PortType (Interfaces)	6.5, 7.2.1	M	PR 4.2.1	
	- Binding (Transport)	7.2.2	M	PR 4.2.2a	

Profile Requirements List (PRL)	NTCIP 2306 Section	Mandatory / Optional	Profile Requirement	Project Requirement
- Service (Transport)	7.2.3	M	PR 4.2.2a	
c) Message Encoding				
SOAP	4.2.2	M	PR 1.2, 4.2.1a, 4.2.1b	
d) Message Transport				
HTTP	5.1.3	M	PR 2.1a, 4.1.2a, 4.2.2a	
HTTPS	5.1.4, 6.6	O	PR 2.1b	
2.0 XML over HTTP				
a) WSDL Request Only (XML Direct)		M		
- WSDL General	6.1	M		
- Definitions	6.2	M	PR 3.1	
- Types/Schema	6.3	M	PR 3.1, 3.2	
- Message	6.4	M	PR 3.3	
- PortType (Interfaces)	8.1.1	M	PR 4.4.1, 3.4	
- Binding (Transport)	8.1.2	M	PR 4.4.2a	
- Service (Transport)	8.3	M	PR 4.4.2a	
b) WSDL Request-Response		O		
- WSDL General	6.1	M		
- Definitions	6.2	M	PR 3.1	
- Types/Schema	6.3	M	PR 3.1, 3.2	
- Message	6.4	M	PR 3.3	
- PortType (Interfaces)	8.2.1	M	PR 4.1.1	
- Binding (Transport)	8.2.2	M	PR 4.1.2b	
- Service (Transport)	8.3	M	PR 4.1.2b	
d) Message Encoding				
XML Text	4.1.2	M	PR 1.1a, 4.2.2b	
XML Gzip	4.1.2	O	PR 1.1 a, 4.2.2b	
e) Message Transport				
HTTP	5.1.1, 5.1.2	M	PR 2.1a, 4.2.2b	
HTTPS	5.1.4, 6.6	O	PR 2.1a, 4.2.2b	
3.0 XML over FTP				
a) WSDL Request Only (XML Direct)		M		
- WSDL General	6.1	M		
- Definitions	6.2	M	PR 3.1	
- Types/Schema	6.3	M	PR 3.1, 3.2	
- Message	6.4	M	PR 3.3	
- PortType (Interfaces)	9.1.2	M	PR 4.4.1, 3.4	

Profile Requirements List (PRL)	NTCIP 2306 Section	Mandatory / Optional	Profile Requirement	Project Requirement
- Binding (Transport)	9.1.3	M	PR 4.4.2b	
- Service (Transport)	9.1.4	M	PR 4.4.2b	
b) Message Encoding (one of the following)				
XML Text	4.1.1	O	PR 1.1a, 4.4.2b	
XML Gzip	4.1.2	O	PR 1.1b, 4.4.2b	
c) Message Transport				
FTP	5.2.1	M	PR 2.2a, 4.4.2b	

SECTION 4 MESSAGE ENCODING

4.1 XML MESSAGE ENCODING

This section provides a normative reference for an XML message.

4.1.1 XML Text Encoding

The following applies to an XML message or file:

Normative

1. The XML message or file shall be an XML version 1.0 document.
2. The XML message shall contain the following header:

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

followed by the message content.

3. The XML message or file shall be capable of being validated against any of the XML schemas referenced in the WSDL.

4.1.2 XML Gzip Encoding

Normative

A message may be encoded using the GZIP format. When such a message is uncompressed, the message payload shall be a single well formed XML message meeting the requirement of clause 4.1.1

4.2 SOAP MESSAGE ENCODING

Informative

The <soap:header> contains application specific information not covered by this standard.

The following represents the correct form of a SOAP messages.

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <!-- message content -->
  </soap:Body>
</soap:Envelope>
```

Normative

1. The SOAP message shall conform with the requirements of the WS-I Basic Profile Section 4.2.6.
2. The SOAP message shall consist of a <soap:Envelope> tag with two internal tags: a <soap:Header> tag followed by a <soap:Body> tag.

3. The <soap:Header> tag is provided to ensure that center-to-center communications software is able to handle a <soap:Header>, as specified by the WS-I Basic Profile, even if no header information is provided.
4. The <soap:Body> open and close tags encapsulate an XML Message that shall be capable of being validated using the XML Schema(s) referenced in the WSDL.

SECTION 5 MESSAGE TRANSPORT

5.1 MESSAGE TRANSPORT USING HTTP

This section and the following sections provide a normative reference for transport of SOAP and XML encoded messages over HTTP.

5.1.1 HTTP Headers for XML Text and Gzip

Normative

1. The HTTP Headers shall conform with the IETF HTTP/1.1 RFC 2616 specification, Section 4.2 - Message Headers.

5.1.2 HTTP Headers for SOAP

Normative

2. The SOAP message shall conform with the requirements of the WS-I Basic Profile Section 3.4.

5.1.3 HTTP Secure Sockets (HTTPS)

Normative

1. The use of Secure Sockets message shall conform with the requirements of the WS-I Basic Profile Section 6.1.

5.1.4 HTTP Version

Normative

1. The HTTP Version shall conform with the requirements of the WS-I Basic Profile Section 3.4.

5.2 MESSAGE TRANSPORT USING FTP

This section and the following sections provide a normative reference for transport of XML encoded files using the FTP.

5.2.1 FTP

Informative

This standard supports only the FTP 'GET'.

Normative

1. The use of FTP shall be conform with the requirements of the IETF FTP RFC 959.

SECTION 6 SUB PROFILES COMMON

6.1 WSDL COMMON

This section and the following sub sections provide a normative reference for WSDL that apply to all of the sub profiles, based on the requirements and solutions identified in the previous sections.

In this section and the sections that follow, the representative examples given have been taken from the TMDD schema. Normative requirements which mention the TMDD schema and namespace should be replaced with similar alternative terms for the other functional areas of ITS (ATIS, IM, TCIP, LRMS, ITIS, DSRC, NTCIP, etc.) as required when this standard is used for those areas.

6.1.1 General Requirements

Informative

It is the prerogative of each data provider to determine which services and operations to support. The use of multiple services by a data provider, a center may determine to specify different services or versions of a service to support users that have different access privileges. In other words, a data provider may choose to expose different interfaces and different sets of supported messages and dialogs to different classes of users. Observe that most of the message set schemas developed by other standard bodies, which expect to use this profile contain multiple messages and dialogs, only a portion of which would be expected to be implemented in any particular deployment.

Normative

1. The version of WSDL shall conform with the requirements of the WS-I Basic Profile 1.1, which is WSDL 1.1.
2. A center that provides a service to one or more external centers, and/or expects the external center to implement a service (e.g., a subscriber's callback listener) shall document the services using the WSDL.
3. A center may provide different WSDL documents for different classes of users.

6.1.2 Required WSDL Sections

Normative

The WSDL shall consists of 6 sections as follows:

1. Definitions
2. Types/Schema
3. Message
4. PortType
5. Binding
6. Service

6.1.3 Profile Common WSDL Sections

Informative

The definitions, types/schemas, and message sections are described in this section and apply across all sub profiles.

6.1.3.1 Description of the Contents of the Definitions Section

The definitions section primarily includes namespace declarations for the entire document.

6.1.3.2 Description of the Contents of the Types/Schema Section

The types section is used to define the format of a message, and is where a referenced XML schema is imported into the WSDL document allowing the WSDL to reference the messages defined in an external XML schema.

6.1.3.3 Description of the Contents of the Message Section

The message section found in the WSDL document lists the messages that will be used by other portions of the document, in essence, defining the list of messages supported by a specific implementation.

Note that this contrasts with the XML schemas used in the typical deployment. Typically the XML schema used in a deployment is taken directly from the standard and is used without change and typically contains additional elements and messages from the standard which are not used in the specific deployment. The WSDL document contains only those message and dialogs which the specific deployment supports, and provide the locally unique paths, ports and bindings used. These messages are defined by the schema cited. Thus, WSDL documents are unique in these respects to each deployment.

6.1.4 Sub Profile Specific WSDL Sections

Informative

The portType, binding, and service section are specific to each sub profile (and moreover, each message transmission pattern described in the sub profile). A brief description of each follows:

6.1.4.1 Description of the Contents of the PortType Sections

The portType section defines the interfaces supported by the center: operations, and associated input and output messages.

The operations defined in the PortType section should be defined by each center, and may be a national, agency, or project-specific. The following precedence is recommended: where no national level specification of dialogs exists, an agency may specify the dialogs, and in the case where no agency-specific dialogs exist, a project should specify the dialogs for their center interfaces.

In either case, an SDO, agency specification, or a project requirements document should specify which messages are related with which center operations (for example, using sequence diagrams to show the necessary dialogs). In the case of TMDD, which has specified dialogs at the national level, the relationship between messages and operations is well understood. An agency or project should simply need to specify which of the superset of dialogs defined at the national level the center(s) need to be supported by centers at the agency (or project level, where no agency specification exists). The messages set in other functional areas follow a similar pattern of use.

6.1.4.2 Description of the Contents of the Binding Sections

The binding section defines the transport protocol over which the messages are transmitted. In the case of this standard: HTTP or FTP.

6.1.4.3 Description of the Contents of the Service Sections

The service section defines the endpoints (a URI location) to which an external (data consumer or data user) can connect to request the service.

6.2 WSDL DEFINITIONS SECTION

Informative

The following represents the correct form of a definitions section of the WSDL, which specifies the Namespaces and Target Namespace for the WSDL. Again, the schema and namespace of TMDD is used here as an example and other functional areas would replace this term when needed.

```
<definitions name="tmddService" targetNamespace="http://www.tmdd-service"
  xmlns:tns="http://www.tmdd-service"
  xmlns:tmdd="http://www.tmdd-address"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:ftp="http://schemas.ntcip.org/wsdl/ftp/"
>
```

Normative

1. The <definitions> tag shall include a name attribute.
2. The targetNamespace shall be specified, for example, "http://www.tmdd-service". - Note that this targetNamespace comprises the common functional abbreviation of the subject area and also a revision string to denote precisely which release of the schema standard is being referenced. It is not a working URL reference, although it is commonly mistaken for one.
3. The xmlns:tns namespace shall be specified and be the same as the targetNamespace.
4. The functional area namespace (here xmlns:tmdd) shall be specified and be the same as that in the referenced functional area XML Schema which provides the message definitions (here TMDD).
5. The following additional schema namespaces shall be specified:
 - a. xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
 - b. xmlns:xs="http://www.w3.org/2001/XMLSchema"
 - c. xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
 - d. xmlns="http://schemas.xmlsoap.org/wsdl/"
 - e. xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
 - f. xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
 - g. xmlns:ftp="http://schemas.ntcip.org/wsdl/ftp/"

6.3 WSDL TYPES AND SCHEMA SECTION

Informative

The following is an example that represents the correct form for import of a schema into a WSDL.

```
<types>
  <schema targetNamespace="http://www.tmdd-service"
    xmlns:tmdd="http://www.tmdd-address"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
  >
```

```
        elementFormDefault="qualified">

        <appinfo><documentation>
            TMDD XML Schema Version 1.5 December 15, 2003

        </documentation></appinfo>

        <xs:import
            namespace="http://www.tmdd-address" schemaLocation="TMDD.xsd"/>
    </schema>
</types>
```

Normative

1. The type section shall begin with the <types> tag.
2. The <schema> tag shall immediately follow the <types> tag.
3. The <schema> targetNamespace attribute shall be the same the targetNamespace of the WSDL definition section.
4. The following namespaces should be specified:
 - a. xmlns:tmdd="http://www.tmdd-address" (matching the namespace of the WSDL definition).
 - b. xmlns="http://www.w3.org/2001/XMLSchema"
 - c. xmlns:xs="http://www.w3.org/2001/XMLSchema"
5. An <appinfo><documentation> tag pair should be used to convey the XML schema version and date as well as any other meta data information that is useful.
6. The <xs:import> tag shall follow an <appinfo><documentation> tag pair and specify the namespace being imported and the schemaLocation, i.e., the name (or URL) of the XML Schema file. In this case, the XML schema file is shown as a relative location from the location of the WSDL file. It is recommended that the XML Schema location be shown as a relative location with respect to the WSDL file location. As the other linked schema files used in ITS also use relative locations to point to each other, this is in keeping with ITS industry conventions.
7. Additional namespaces may be included that are project specific.

6.4 WSDL MESSAGE SECTIONS

Informative

The following example represents the correct form for specifying a single part message.

```
<message name="MSG_DMSSControlRequest">
    <part name="message" element="tmdd:dMSSControlRequest"/>
</message>
```

Normative

1. The message section shall specify all top-level messages defined in the XML Schema that apply to the project implementation.
2. The message name shall begin with the prefix "MSG_" followed by the name of the message type as specified in the XML Schema.
3. The element part of the part name shall use the form "tmdd:" followed by the name of the XML Schema element being imported into the WSDL.

6.5 DESCRIPTION OF THE PUBLICATION-SUBSCRIPTION MESSAGE TRANSMISSION PATTERNS

6.5.1 Subscription

Informative

The concept of a Subscriber Callback Listeners is described in this standard (see Section 7.2).

The subscription, implemented as a request-response message transmission pattern, can be used for both one time replies and on-going subscriptions. SDOs or implementations developing or extending message sets that will be used in conjunction with this standard should refer to clause 6.5.3 for further information.

Normative Responsibilities of the Subscriber

1. The subscription shall be implemented (initiated) as a request-response message transmission pattern.
2. The subscription information shall be included in the "request" message of the subscriber.
3. The subscriber shall implement a Subscriber Callback Listener to receive publication messages.
4. The subscriber center shall report an error to the publisher as an HTTP return code.
5. A subscription message shall include one or more XML tags or a complete XML fragment which shall supply the following data concepts in some well established format :
 - returnAddress. This shall be a URI that defines either a SOAP over HTTP endpoint, or an XML over HTTP "POST" endpoint. This will allow the publication center to send a message over HTTP.
 - subscriptionAction. The subscriptionAction includes the following:
 - 1) Create New Subscription
 - 2) Update Replace a Subscription (restarts a subscription)
 - 3) Cancel a Subscription
 - 4) Cancel All Prior Subscriptions (subscription id is ignored)
 - subscriptionID. A unique ID within the subscriber's center. The publication center may add another unique ID to fully define a given publication/subscription stream.

6.5.2 Publication

Normative Responsibilities of the Publisher

1. The WSDL (created by the publication center) shall show that subscription and publication use the same XML encoding and transmission message patterns (SOAP over HTTP or XML over HTTP).
2. The publication center shall specify the subscriber callback listener WSDL.
3. The publication shall be implemented as a request-response transmission pattern.
4. The publication information shall be included in the "request" message (from publication center to subscriber center).
5. The publication center shall report an error to the subscriber as an HTTP return code.
6. A publication message shall include one or more XML tags or a complete XML fragment for the following:
 - subscriptionCount. Each published message sent to a subscriber shall include a number starting with 1 through 65,535 ($2^{16}-1$) and which increments by 1 with each new message which identifies the message sequence. This will allow a subscriber to know when it has missed a message. It is the responsibility of the subscriber to restart a subscription upon receiving knowledge of a missed message, if required. Once the count reaches 65,535, the count begins with 1 again.

- subscriptionID. This shall be the subscriber center's subscription ID. Taken together with the subscriber (or some form of ID) it forms a unique value. More than one subscription ID per subscriber is supported by most message sets, however this is not a requirement in deployment.
7. A publication center may implement the following: upon receiving a knowledge that a message has been missed, a subscriber may request that the missing message be resent.

6.5.3 Responsibilities of the Message Set SDOs

Informative

It is the responsibility of the SDOs defining the message sets to define a message to allow a subscriber to register for a subscription. If a SDO that defines a message set decides to reference this standard to support the publish-subscribe mechanism described, then the following section applies.

The normative reference below is an analysis and synthesis of the approaches taken by the IEEE 1512 and SAE ATIS message sets, and the common themes shared as relate to publish and subscribe. It is expected that the message set standards will define a publication and subscription messages, data frames, data elements, and optional features, which in turn will be implemented in a local deployment.

This standard focuses on the definition of a center's interface that implements publish and subscribe message transmission pattern (written as WSDL), the message encoding rules that apply (SOAP or XML), and message transport (HTTP).

6.6 DESCRIPTION OF TRANSPORT FOR SECURE SOCKET SERVICE ENDPOINTS

The following apply to specify the use of secure sockets in conjunction with the HTTP:

- A HTTPS service endpoint shall be specified by using "https:" instead of "http:" as the transport protocol.

6.7 SPECIFYING ADDITIONAL WEB SERVICE ITEMS

The WSDL may contain additional schemas, types, and messages that are project specific ONLY if there is no other XML schema, type, or message available that meet the project requirements.

SECTION 7 WSDL FOR SUB PROFILE SOAP OVER HTTP

7.1 WSDL FOR SOAP REQUEST / RESPONSE

7.1.1 WSDL for SOAP Request-Response PortType Section

Informative

The following example represents the correct form for specifying a simple Request-Response Port Type.

```
<!-- Request/Response Port Type Operation -->
<portType name="tmddServiceSOAPPort">
  <operation name="OP_RequestEventInformation">
    <input message="tns:MSG_EventFilterRequest"/>
    <output message="tns:MSG_FullEventUpdateAndActionLogResponse"/>
  </operation>

  <operation name="OP_ShareVSControl">
    <input message="tns:MSG_CCTVSwitchCommandRequest"/>
    <output message="tns:MSG_CCTVSwitchCommandResponse"/>
  </operation>
</portType>
```

Normative

1. The <portType> tag shall be the parent tag for a list of operations supported by the SOAP Service for the specific project implementation.
2. The operation name shall begin with the prefix "OP_" followed by a descriptive name for the operation.
3. Each operation shall have one <input> and one <output> tag. The <input> tag is specified first followed by the <output> tag.
4. The input message shall reference a message defined in the Message section of the WSDL. To reference a message inside the WSDL, the message part of the operation shall begin with the prefix "tns:" followed by the name of the message specified in the Message section of the WSDL.

7.1.2 WSDL for SOAP Request-Response Binding Section

Informative

The following represents the correct form for specifying a SOAP over HTTP binding. Please note the adherence to the document/literal form.

```
<binding name="tmddServiceSOAPBinding" type="tns:tmddServiceSOAPPort">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="OP_RequestEventInformation">
    <soap:operation soapAction="OP_RequestEventInformation" style="document"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
```

```
</operation>  
</binding>
```

Normative

1. The <binding> tag shall be followed by a <soap:binding> tag.
2. The style attribute of the <soap:binding> tag shall be "document"
3. The transport attribute of the <soap:binding> tag shall be "http://schemas.xmlsoap.org/soap/http"
4. The operation name must match the operation as specified in the Port Type section of the WSDL for each operation supported by the center.
5. The <operation> tag shall be followed by the <soap:operation> tag.
6. The soapAction attribute specifies the soapAction which the external center requester must include an HTTP header. There is no requirement for a soapAction, however, if no soapAction is specified, the soapAction attribute must be written as two consecutive double quote characters ("").
7. The <soap:operation> tag shall be followed by a <input> </input> tag set.
8. The <input> tag shall be followed by a <soap:body> tag. The use attribute of the <soap:body> tag shall be "literal". This specifies that the message content of the SOAP message is an XML message that conforms with the XML Schema referenced in the schema section of the WSDL, which in turn is enclosed between a <soap:envelope> and <soap:body> tags.
9. The <input> tag set shall be followed by an <output> tag set.
10. The <output> tag shall be followed by a <soap:body> tag. The use attribute of the <soap:body> tag shall be "literal".

7.1.3 WSDL for SOAP Request-Response Service Section

Informative

The WSDL Service sections specify the endpoints that will implement the services defined. The following example represents the correct form for specifying a SOAP Service.

```
<service name="tmddSOAPService">  
  <documentation>Traffic Management Service</documentation>  
  <!-- connect it to the binding "tmddServiceSOAPBinding" above -->  
  <port name="tmddServiceSOAPPort" binding="tns:tmddServiceSOAPBinding">  
    <!-- give the binding an network address -->  
    <soap:address  
      location="http://www.mycenter.org/c2cxml/tmdd/tmddSOAPService"/>  
  </port>  
</service>
```

Normative

1. The <service> tag shall contain a name attribute of the service.
2. The <service> tag may be followed by a <documentation>tag that contains a brief description. If a <documentation> tag exists, it should be placed directly after the <service> tag.
3. The <port> tag shall follow the <documentation> tag, if one is present. The name attribute of the name tag will reflect the name of a SOAP port followed by the binding attribute indicating the name of the binding identified in the bindings section which applies.
4. Following the <port> tag, a <soap:address> tag shall be placed. The location attribute of the <soap:address> tag shall specify the service endpoint of the SOAP service.
5. The service endpoint must specify a valid URL.
6. Multiple instances of the <port> element may be defined as needed.

7.2 WSDL FOR SOAP SUBSCRIBER'S CALLBACK LISTENER (PUBLISH-SUBSCRIBE)

This section provides for the specification of WSDL for subscribers necessary to implement a listener (data consumer) for publications from a server (data provider).

7.2.1 WSDL for SOAP Subscriber's Callback Listener PortType Section

Informative

The publication center defines the WSDL necessary to specify how to receive a callback message. The following example represents the correct form for specifying a subscriber callback Port Type.

```
<portType name="tmddServiceSubscriberCallbackSOAPPort">
  <operation name="OP_SubscriberEventUpdateInformation">
    <input message="tns:MSG_BasicEventUpdate"/>
    <output message="tns:MSG_Acknowledge"/>
  </operation>
</portType>
```

Normative

1. The <portType> tag shall be followed by a list of operations supported by the subscriber for listening for asynchronous SOAP message delivery.
2. The list of operations a subscriber listener must support shall be defined by the publishing center and documented in the publication center's WSDL.
3. The operation name, input message, and output message follow the same rules as those defined for a SOAP Request-Response PortType defined in clause 7.1.3.
4. The callback message (sent from the publisher to subscriber) shall be defined as an "input message" (from the subscriber's perspective).
5. The subscriber shall respond with an acknowledge message (if defined) or an HTTP error code, which shall be defined in the "output message".

7.2.2 WSDL for SOAP Subscriber's Callback Listener Binding Section

Informative

The following represents the correct form for specifying a SOAP binding for a subscriber's callback listener. This is consistent with the concept that a subscriber (external center acting as a data consumer) will provide a SOAP service to listen for the asynchronous messages sent from the center.

```
<binding name="tmddServiceSubscriberCallbackSOAPBinding"
type="tns:tmddServiceSubscriberCallbackSOAPPort">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="OP_SubscriberEventUpdateInformation">
    <soap:operation soapAction="OP_SubscriberEventUpdate" style="document"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  <!-- continue with each of the operations a subscriber callback listener must support -->
```

</binding>

Normative

1. The normative rules from the SOAP Request-Response Binding defined in clause 7.1.3 shall apply.

7.2.3 WSDL for SOAP Subscriber's Callback Listener Service Section

Informative

The following example represents the correct form for specifying a SOAP Subscriber's Callback Service. The SOAP Subscriber's Callback Service specifies a template that an external center will use, along with the appropriate binding, portType, messages and schema defined, to define a callback listener.

```
<service name="tmddSOAPServiceSubscriberCallback">
  <documentation>Subscribers of my Traffic Management Service
    </documentation>
  <!-- connect it to the binding "tmddServiceSubscriberCallbackSOAPBinding" above -->
  <port name="tmddServiceSubscriberCallbackSOAPPort"
    binding="tns:tmddServiceSubscriberCallbackSOAPBinding">
    <!-- give the binding an network address -->
    <soap:address location="http://www.tmddservice-subscriber-callback"/>
  </port>
</service>
```

Normative

1. The <service> tag shall contain a name attribute of the service.
2. The <service> tag may be followed by a <documentation> tag that contains a brief description. If a <documentation> tag exists, it should be placed directly after the <service> tag.
3. The <port> tag shall follow the <documentation> tag, if one is present. The name attribute of the name tag will reflect the name of a SOAP port followed by the binding attribute indicating the name of the binding identified in the bindings section which applies.
4. Following the <port> tag, a <soap:address> tag shall be placed. The location attribute of the <soap:address> tag shall specify a fake (or dummy) service endpoint of the SOAP service. This service endpoint of the external center's subscriber callback listener may either be registered manually or at run time. However, to support valid WSDL, the <soap:address> tag must be included.

SECTION 8 WSDL FOR SUB PROFILE XML OVER HTTP

8.1 WSDL FOR XML OVER HTTP (REQUEST ONLY FILE RETRIEVAL)

8.1.1 WSDL for XML Over HTTP Request Only PortType Section

Informative

The following represents the correct form for specifying a request-only HTTP “GET” operation for specifying access to XML content files or messages via HTTP.

```
<portType name="tmddXMLHTTPPort">
  <operation name="OP_PublishAllDeviceStatusInformation">
    <documentation>
      updated every 5 minutes</documentation>
    <input message="tns:MSG_AllDeviceStatus"/>
  </operation>
</portType>
```

Normative

1. The WSDL to describe the simple retrieval of a file shall use a one-way operation to maintain consistency with the WS-I Basic Profile and other portions of this standard.
2. The <portType> tag shall be followed by a single operation that maps to a specific file.
3. The operation shall be a one-way operation between the data provider center and the external center (data consumer).
4. The operation name shall begin with the prefix “OP_” followed by a descriptive name for the operation.
5. Each operation shall have one <input> tag.
6. The operation may also contain a <documentation> tag which shall be used to specify the frequency of update of the XML file as well as other useful meta data

8.1.2 WSDL for XML Over HTTP Request Only Binding Section

Informative

The XML over HTTP request only binding uses the “GET” form of the HTTP.

After the <http:binding> tag comes a list of operations, with each operation bound to a file via the location attribute. The complete URL of the file is the concatenation of the location of the XMLHTTP Service (see Specifying Service Section, below) with the file name specifying in the location attribute of the <operation> tag.

For example, if the XMLDirect Service location attribute is specified as *http://www.mycenter.org* and the binding operation location is specified as */CCTVDeviceStatus.xml*, then the URL for the file is *http://www.mycenter.org/CCTVDeviceStatus.xml*.

The following example represents the correct form for specifying the binding of operations (as listed in the Port Type section) to and HTTP “GET”. The example represents the correct form for specifying the binding for an XML document over HTTP. The output may be any MIME type as specified in the WS-I Basic Profile 1.1, including: “gzip” for zip-encoded compressed data, “text/xml” for an XML Document, and “image/jpeg” or “image/gif” for image data.

```
<binding name="tmddXMLHTTPGetBinding" type="tns:tmddXMLHTTPGetPort">
  <http:binding verb="GET"/>
```

```
<!-- All Device Inventory-->
<operation name="OP_PublishAllDeviceInventory">
  <http:operation location="/AllDeviceInventory.xml"/>
  <input>
    <http:urlEncoded/>
  </input>
  <output>
    <mime:content type="text/xml"/>
  </output>
</operation>
</binding>
```

Normative

1. The <binding> tag shall be followed by a <http:binding> tag.
2. The http:binding verb attribute shall be "GET".
3. The operation name shall match the operation as specified in the Port Type section of the WSDL for each operation (file) supported by the center.
4. The <operation> tag shall be followed by the <http:operation> tag.
5. The location attribute shall be the name of the file. The name of the file is to be defined by each center. However, it is recommended that the file name reflect the name of the message or messages contained. Documentation outlining these naming rules should be established and a link provided to it in the meta-data of the <documentation> section.
6. The <http:operation> tag shall be followed by the <input> </input> tag set.
7. The <input> tag shall be followed by a <http:urlEncoded> tag.
8. The <input> tag set shall be followed by an <output> </output> tag set.
9. The <output> tag shall be followed by a <mime:content> tag.
10. The valid type attribute of the <mime:content> tag shall any MIME type specified in the WS-I Basic Profile and supported by the center server.
11. If the MIME type attribute is "gzip", then the uncompressed information must be of type "text/xml".

8.2 WSDL FOR XML OVER HTTP (REQUEST-RESPONSE)

The section describes the WSDL to support use of HTTP "POST". The WSDL fragments show only one POST argument shall be sent over HTTP. The single POST argument shall contain an XML message in text form. This approach will allow implementation based on CGI and web server scripting such as PERL, ASP, and PHP, what have you).

8.2.1 WSDL for XML Over HTTP Request-Response PortType Section

The PortType section for XML over HTTP follows the same rules as those in Section 7.1.1 - WSDL for SOAP Request-Response PortType Section.

8.2.2 WSDL for XML Over HTTP Request-Response Binding Section

Informative

The following example represents the correct form for specifying the binding of operations (as listed in the Port Type section) to and HTTP "POST". The example represents the correct form for specifying the binding for an XML document over HTTP.

```
<binding name="tmddXMLHTTPPostBinding" type="tms:tmddXMLHTTPPostPort">
  <http:binding verb="POST"/>
```

```
<operation name="OP_PublishAllDeviceInventory">
  <http:operation location="/AllDeviceInventory.xml"/>
  <input>
    <mime:content type="application/x-www-form-urlencoded"/>
  </input>
  <output>
    <mime:content type="text/xml"/>
  </output>
</operation>
</binding>
```

Normative

1. The same requirement of the Request Only Binding shall apply.
2. The content of the <input> tag shall be ONE and ONLY ONE form parameter, which shall contain an XML message.

8.3 WSDL FOR XML OVER HTTP SERVICE SECTION

Informative

The following represents the correct form for specifying an XML instance over HTTP Service.

```
<service name="tmddXMLHTTPService">
  <documentation> Traffic Management XML over HTTP Service
  </documentation>

  <port name="tmddXMLHTTPPort" binding="tns:tmddXMLHTTPGetBinding">
    <http:address location="http://www.mycenter.org/c2cxml/tmdd"/>
  </port>

  <port name="tmddXMLHTTPPort" binding="tns:tmddXMLHTTPPostBinding">
    <http:address location="http://www.mycenter.org/c2cxml/tmdd"/>
  </port>

</service>
```

Normative

The rules that apply are the same as those specified for a SOAP Service with the following exception:

1. The XML HTTP endpoint uses the <http:address> tag to specify an endpoint rather than the <soap:address> tag.
2. Also, please note that the actual location of the XML file is a concatenation of the location attribute specified in the <http:address> (of the service) with the location attribute of the <http:operation> (of the binding).

SECTION 9 WSDL FOR SUB PROFILE XML OVER FTP

9.1 WSDL FOR XML OVER FTP (FILE RETIEVAL)

This section for use of XML over FTP is modeled after Section 8.1 WSDL for XML Over HTTP (Request Only File Retrieval) which uses the HTTP "GET".

9.1.1 Extending WSDL for FTP

At the time of this writing, the WSDL specification does not specify an FTP binding. However, since both HTTP and FTP support a 'GET' operation, this standard specifies the creation of an FTP XML namespace, and WSDL syntax very similar to that used in the HTTP 'GET'.

9.1.2 WSDL for XML over FTP PortType Section

The following represents the correct form for specifying access to XML content files via FTP.

```
<portType name="tmddXMLFTPPort">
```

The <portType> tag is followed by each operation which is related to a specific file. The operation is a one-way operation between the center and the external center. (Please note that neither the operation nor message below is defined within TMDD.)

The following represents the correct form for specifying a one-way operation.

```
<operation name="OP_PublishAllDeviceStatusInformation">  
  <documentation>updated every 5 minutes</documentation>  
  <input message="tns:MSG_AllDeviceStatus"/>  
</operation>
```

The following applies:

1. The operation name and output message follow the same rules as those defined for a SOAP Service Port Type.
2. The operation may also contain a <documentation> tag which should be used to specify the frequency of update of the XML file.

9.1.3 WSDL for XML over FTP Binding Section

The following represents the correct form for specifying the binding for XML over FTP.

```
<binding name="tmddXMLFTPBinding" type="tns:tmddXMLFTPPort">  
  <ftp:binding verb="GET"/>
```

The <binding> tag is followed by a <ftp:binding> tag, for which the following applies:

1. The verb attribute shall be "GET"

After the <ftp:binding> tag comes a list of operations, with each operation bound to a file via the location attribute. The complete URL of the file is the concatenation of the location of the XMLFTP Service (see Specifying Service Section, below) with the file name specifying in the location attribute of the <operation> tag.

For example, if the XML FTP Service location attribute is specified as *http://www.mycenter.org* and the binding operation location is specified as */CCTVDeviceStatus.xml*, then the URL for the file is *http://www.mycenter.org/CCTVDeviceStatus.xml*.

The following represents the correct form for specifying the binding of operations (as listed in the Port Type section) to FTP.

```
<!-- All Device Inventory-->
<operation name="OP_PublishAllDeviceInventory">
  <ftp:operation location="/AllDeviceInventory.zip"/>
  <input>
    <ftp:urlEncoded/> <!-- FILE NAME -->
  </input>
  <output>
    <mime:content type="gzip"/>
  </output>
</operation>
```

The following applies:

1. The operation name must match the operation as specified in the Port Type section of the WSDL for each operation (file) supported by the center.
2. The <operation> tag shall be followed by the <ftp:operation> tag.
3. The location attribute shall be the name of the file. The name of the file is to be defined by each center. However, it is recommended that the file name reflect the name of the message or messages contained.
4. The <ftp:operation> tag shall be followed by the <input> </input> tag set.
5. The <input> tag shall be followed by a <ftp:urlEncoded> tag.
6. The <input> tag set shall be followed by an <output> </output> tag set.
7. The <output> tag shall be followed by a <mime:content> tag. The valid type attribute of the <mime:content> tag shall be MIME type supported by the center server (assuming that the center MIME types are also valid per the FTP specification). Example MIME types include: "gzip" for zip-encoded compressed data, "text/xml" for an XML Document, and "image/jpeg" or "image/gif" for image data. If the "gzip" MIME type is included, the compressed information must be XML.

9.1.4 WSDL for XML over FTP Service Section

The following represents the correct form for specifying an XML FTP Service.

```
<service name="tmddXMLFTPService">
  <documentation>
    Traffic Management XML FTP Service
  </documentation>
  <!-- connect each port to a specific file - a pseudo end-point -->

  <port name="tmddXMLFTPPort" binding="tns:tmddXMLFTPBinding">
    <ftp:address location="ftp://www.mycenter.org/c2cxml/tmdd"/>
  </port>
</service>
```

The rules that apply are the same as those specified for a SOAP Service with the following exception:

1. The XML FTP endpoint uses the <ftp:address> tag to specify an endpoint rather than the <soap:address> tag.
2. Insert comment about documentation again here
3. Also, please note that the actual location of the XML file is a concatenation of the location attribute specified in the <ftp:address> (of the service) with the location attribute of the <ftp:operation> (of the binding).
4. The address shall be a valid URL.

SECTION 10 TEST PLAN DEVELOPMENT AND CONFORMANCE GUIDANCE

10.1 WSDL DEVELOPMENT METHODOLOGY

A fully elaborated WSDL sample, based on the TMDD dialogs document referenced by this interim standard, is included in Appendix B.

The following steps were taken to create the WSDL contained in Appendix B.

1. The initial step was to document the operation, input messages, and output messages of each dialog in table form. (See Appendix A).
2. From a reading of the documentation included in the TMDD and by interpreting the associated dialog, the dialog was characterized as either: Request-Response, Subscribe/Callback, or One-Way.
3. Finally, a judgement was made as to which dialogs would be reasonable to support using the XMLDirect method.
4. The contents of steps 1-3 are documented in tabular form for each dialog contained in the TMDD dialogs, and shown in Appendix A. Appendix A was then used as a basis for development of the TMDD Profile needs listed in Section 2 of this document.
5. Finally, a simple straightforward translation of the table shown in Appendix A table to WSDL was accomplished, and the WSDL checked using a WSDL syntax checker.

It is recommended that a similar process take place for development of agency- or project- specific WSDL.

[If (and a big If at that) we get formal funding for add the auto WSDL to Mini-Edit we might mention it here as an alternative which will have produced many of the mappings but which will still need the user to select which items they want, then provide various paths and mapping endpoints to create local WSDL.]

10.2 TESTING THE CORRECTNESS OF THE WSDL

WSDL correctness may be inspected in several ways. The first is to use a WSDL syntax checker. Several of which are available on the internet. A second way to check the WSDL syntax is to validate the WSDL against the WSDL schema (i.e., the schema that specifies what WSDL should look like).

NTCIP 2306 v01.51

with shortened Appendices A & B (WSDL Worksheet and Example WSDL)

APPENDICES

Appendix A - Sample WSDL Worksheet	1
Appendix B - Sample WSDL	2
Appendix C - NTCIP C2C XML Profile Mapping to WS-I Basic Profile	7

APPENDIX A - SAMPLE WSDL WORKSHEET

Table A-1. Sample WSDL Worksheet

Service	Operation	Message Input	Message Output	Message Pattern	Message Encoding	Message Transport
tmddSOAPService	OP_ShareDMSControl	MSG_DMSControlRequest	MSG_DMSControlResponse	R/R	SOAP	HTTP
tmddSOAPService	OP_ShareDMSInventoryInformation	MSG_DMSInventoryRequest	MSG_DMSInventory	R/R	SOAP	HTTP
tmddSOAPService	OP_ShareDMSStatusInformation	MSG_DMSStatusRequest	MSG_DMSDeviceStatus	R/R	SOAP	HTTP
tmddSOAPSubscriberCallback	OP_SubscriberCallbackDMSInventoryInformation	MSG_Acknowledge	MSG_DMSInventory	S/CB	SOAP	HTTP
tmddSOAPSubscriberCallback	OP_SubscriberCallbackDMSStatusInformation	MSG_Acknowledge	MSG_DMSDeviceStatus	S/CB	SOAP	HTTP
tmddXMLDirect	OP_PublishDMSInventoryInformation		MSG_DMSInventory	Oneway	XML	HTTP+FTP
tmddXMLDirect	OP_PublishDMSStatusInformation		MSG_DMSDeviceStatus	Oneway	XML	HTTP+FTP

Message Patterns:

R/R = Request-Response

S/CB = Subscription-Callback

1-Way = One way

APPENDIX B - SAMPLE WSDL

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

<definitions name="tmddService" targetNamespace="http://www.tmdd-service"
  xmlns:tns="http://www.tmdd-service"
  xmlns:tmdd="http://www.tmdd-address"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:ftp="http://schemas.ntcip.org/wsdl/ftp/"
>

<!-- TYPES -->

  <types>
    <schema targetNamespace="http://www.tmdd-service"
      xmlns:tmdd="http://www.tmdd-address"
      xmlns="http://www.w3.org/2001/XMLSchema"
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
      elementFormDefault="qualified">

      <xs:import namespace="http://www.tmdd-address" schemaLocation="TMDD.xsd"/>

    </schema>
  </types>

<!-- MESSAGES -->

  <!-- The information below was derived from the XML Schema specified above -->

  <!-- An Acknowledge Message -->
  <!-- This may be added to the TMDD Schema -->
  <message name="MSG_Acknowledge">
    <part name="message" type="xs:string"/>
  </message>

  <!-- Descriptive Name: DMSControlRequest -->
  <message name="MSG_DMSControlRequest">
    <part name="message" element="tmdd:dMSControlRequest"/>
  </message>

  <!-- Descriptive Name: DMSControlResponse -->
  <message name="MSG_DMSControlResponse">
    <part name="message" element="tmdd:dMSControlResponse"/>
  </message>

  <!-- Descriptive Name: DMSDeviceStatus -->
  <message name="MSG_DMSDeviceStatus">
    <part name="message" element="tmdd:dMSDeviceStatus"/>
  </message>

  <!-- Descriptive Name: DMSInventory -->
  <message name="MSG_DMSInventory">
    <part name="message" element="tmdd:dMSInventory"/>
  </message>

  <!-- Descriptive Name: DMSInventoryRequest -->
```

```

<message name="MSG_DMSInventoryRequest">
  <part name="message" element="tmdd:dMSInventoryRequest"/>
</message>

<!-- Descriptive Name: DMSStatusRequest -->
<message name="MSG_DMSStatusRequest">
  <part name="message" element="tmdd:dMSStatusRequest"/>
</message>

<!-- ***** -->
<!-- TMDD Service -->

<!-- tmddService - PORT TYPE OPERATION INPUT / OUTPUTS SOAP PORT -->
<portType name="tmddServiceSOAPPortType">

  <!-- DMS -->
  <operation name="OP_ShareDMSInventoryInformation">
    <input message="tns:MSG_DMSInventoryRequest"/>
    <output message="tns:MSG_DMSInventory"/>
  </operation>

  <operation name="OP_ShareDMSStatusInformation">
    <input message="tns:MSG_DMSStatusRequest"/>
    <output message="tns:MSG_DMSDeviceStatus"/>
  </operation>

  <operation name="OP_ShareDMSControl">
    <input message="tns:MSG_DMSControlRequest"/>
    <output message="tns:MSG_DMSControlResponse"/>
  </operation>

</portType>

<!-- BINDING - OPERATION INPUT / OUTPUTS - SOAP -->
<binding name="tmddServicesSOAPBinding" type="tns:tmddServiceSOAPPortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

  <!-- DMS -->
  <operation name="OP_ShareDMSInventoryInformation">
    <soap:operation soapAction="OP_ShareDMSInventoryInformation"
      style="document"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>

  <operation name="OP_ShareDMSStatusInformation">
    <soap:operation soapAction="OP_ShareDMSStatusInformation"
      style="document"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>

  <operation name="OP_ShareDMSControl">
    <soap:operation soapAction="OP_ShareDMSControl" style="document"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>

```

```

</binding>

<service name="tmddServiceSOAP">
  <documentation>Traffic Management Service</documentation>
  <!-- connect it to the binding "tmddServiceSOAPBinding" above -->
  <port name="tmddServiceSOAPPort" binding="tns:tmddServiceSOAPBinding">
    <!-- give the binding an network address -->
    <soap:address
      location="http://www.mycenter.org/c2cxml/tmdd/tmddSOAPService"/>
  </port>
</service>

<!-- ***** -->
<!-- TMDD Service Subscriber Callback -->

<!-- tmddServiceClientCallbackSOAPPort -->
<!-- PORT TYPE OPERATION INPUT / OUTPUTS FOR A SOAP PORT LISTENER -->
<portType name="tmddServiceSubscriberCallbackSOAPPortType">

  <!-- DMS-->
  <operation name="OP_SubscriberDMSInventoryInformation">
    <input message="tns:MSG_DMSInventory"/>
    <output message="tns:MSG_Acknowledge"/>
  </operation>

  <operation name="OP_SubscriberDMSStatusInformation">
    <input message="tns:MSG_DMSDeviceStatus"/>
    <output message="tns:MSG_Acknowledge"/>
  </operation>

</portType>

<binding name="tmddServiceSubscriberCallbackSOAPBinding"
type="tns:tmddServiceClientCallbackSOAPPortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

  <operation name="OP_SubscriberDMSInventoryInformation">
    <soap:operation soapAction="OP_SubscriberDMSInventoryInformation"
style="document"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>

  <operation name="OP_SubscriberDMSStatusInformation">
    <soap:operation soapAction="OP_SubscriberDMSStatusInformation"
style="document"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>

<service name="tmddServiceSubscriberCallback">
  <documentation>Clients for my Traffic Management Service</documentation>
  <!-- connect it to the binding "tmddServiceClientCallbackSOAPBinding" above -->
  <port name="tmddServiceSubscriberCallbackSOAPPort"
binding="tns:tmddServiceSubscriberCallbackSOAPBinding">
    <!-- give the binding an network address -->
    <soap:address location="http://www.tmddservice-subscriber-callback"/>
  </port>
</service>

<!-- ***** -->

```

```

<!-- TMDD XMLDirect -->

    <!-- XMLDirect Support -->
    <!-- PORT TYPE OPERATION OUTPUTS HTTP PORT -->

    <portType name="tmddXMLDirectPort">

        <!-- DMS-->
        <operation name="OP_PublishDMSInventoryInformation">
            <input message="tns:MSG_DMSInventory"/>
        </operation>

        <operation name="OP_PublishDMSStatusInformation">
            <input message="tns:MSG_DMSDeviceStatus"/>
        </operation>

    </portType>

    <!-- XMLDirect HTTP Binding -->

    <binding name="tmddXMLDirectHTTPBinding" type="tns:tmddXMLDirectPortType">
        <http:binding verb="GET"/>

        <!-- DMS-->
        <operation name="OP_PublishDMSInventoryInformation">
            <http:operation location="/DMSInventoryInformation.xml"/>
            <input>
                <http:urlEncoded/>
            </input>
            <output>
                <mime:content type="text/xml"/>
            </output>
        </operation>

        <operation name="OP_PublishDMSStatusInformation">
            <http:operation location="/DMSStatusInformation.xml"/>
            <input>
                <http:urlEncoded/>
            </input>
            <output>
                <mime:content type="text/xml"/>
            </output>
        </operation>

    </binding>

    <!-- XMLDirect FTP Binding -->

    <binding name="tmddXMLDirectFTPBinding" type="tns:tmddXMLDirectPortType">
        <ftp:binding verb="GET"/>

        <operation name="OP_PublishDMSStatusInformation2">
            <http:operation location="/DMSStatusInformation.xml"/>
            <input>
                <ftp:urlEncoded/>
            </input>
            <output>
                <mime:content type="text/xml"/>
            </output>
        </operation>

    </binding>

    <service name="tmddXMLDirectService">
        <documentation>Traffic Management XMLDirect Service</documentation>
        <!-- connect each port to a specific file - pseudo end-point -->

```

```
<port name="tmddXMLDirectHTTPPort" binding="tns:tmddXMLDirectHTTPBinding">
  <http:address location="http://www.mycenter.org/c2cxml/tmdd"/>
</port>

<port name="tmddXMLDirectFTPPort" binding="tns:tmddXMLDirectFTPBinding">
  <ftp:address location="ftp://www.mycenter.org/c2cxml/tmdd"/>
</port>
</service>
</definitions>
```

APPENDIX C - NTCIP C2C XML PROFILE MAPPING TO WS-I BASIC PROFILE

Table C-1. NTCIP C2C XML Profile Mapping to WS-I Basic Profile 1.1

Rqmt #	Requirement Text	Accepted	Comments / Restrictions	Requirement Type
R0001	Either an INSTANCE's WSDL 1.1 description, its UDDI binding template, or both MUST be available to an authorized consumer upon request.	y	Restricted to WSDL 1.1 service description.	1. conformance
R0002	A DESCRIPTION MAY contain conformance claims regarding instances, as specified in the conformance claim schema.	n	The conformance claim is sent as a soap header. See R0005.	1. conformance
R0003	A DESCRIPTION's conformance claims MUST be children of the wsdl:documentation element of each of the elements: wsdl:port, wsdl:binding, wsdl:portType, wsdl:operation (as a child element of wsdl:portType but not of wsdl:binding) and wsdl:message.	n	See R0005.	1. conformance
R0004	A MESSAGE MAY contain conformance claims, as specified in the conformance claim schema.	n	See R0005.	1. conformance
R0005	A MESSAGE's conformance claims MUST be carried as SOAP header blocks.	n	Conformance claims will not be used as they utilize SOAP Headers.	1. conformance
R0006	A MESSAGE MAY contain conformance claims for more than one profile.	n	See R0005.	1. conformance
R0007	A SENDER MUST NOT use the soap:mustUnderstand attribute when sending a SOAP header block containing a conformance claim.	n	See R0005.	1. conformance
R1000	When an ENVELOPE is a Fault, the soap:Fault element MUST NOT have element children other than faultcode, faultstring, faultactor and detail.	y		2. messaging
R1001	When an ENVELOPE is a Fault, the element children of the soap:Fault element MUST be unqualified.	y		2. messaging
R1002	A RECEIVER MUST accept faults that have any number of elements, including zero, appearing as children of the detail element. Such children can be qualified or unqualified.	y		2. messaging

Rqmt #	Requirement Text	Accepted	Comments / Restrictions	Requirement Type
R1003	A RECEIVER MUST accept faults that have any number of qualified or unqualified attributes, including zero, appearing on the detail element. The namespace of qualified attributes can be anything other than "http://schemas.xmlsoap.org/soap/envelope/".	y		2. messaging
R1004	When an ENVELOPE contains a faultcode element, the content of that element SHOULD ... authority (in that order of preference).	y		2. messaging
R1005	An ENVELOPE MUST NOT contain soap:encodingStyle attributes on any of the elements whose namespace name is "http://schemas.xmlsoap.org/soap/envelope/".	y		2. messaging
R1006	An ENVELOPE MUST NOT contain soap:encodingStyle attributes on any element that is a child of soap:Body.	y		2. messaging
R1007	An ENVELOPE described in an rpc-literal binding MUST NOT contain soap:encodingStyle attribute on any element that is a grandchild of soap:Body.	n	C2C basic profile will not use RPC.	2. messaging
R1008	An ENVELOPE MUST NOT contain a Document Type Declaration.	y		2. messaging
R1009	An ENVELOPE MUST NOT contain Processing Instructions.	y		2. messaging
R1010	A RECEIVER MUST accept messages that contain an XML Declaration.	y		2. messaging
R1011	An ENVELOPE MUST NOT have any element children of soap:Envelope following the soap:Body element.	y		2. messaging
R1012	A MESSAGE MUST be serialized as either UTF-8 or UTF-16.	y	Restricted to UTF-8.	2. messaging
R1013	An ENVELOPE containing a soap:mustUnderstand attribute MUST only use the lexical forms "0" and "1".	y		2. messaging
R1014	The children of the soap:Body element in an ENVELOPE MUST be namespace qualified.	y		2. messaging
R1015	A RECEIVER MUST generate a fault if they encounter an envelope whose document element is not soap:Envelope.	y		2. messaging
R1016	A RECEIVER MUST accept fault messages that carry an xml:lang attribute on the faultstring element.	y		2. messaging
R1017	A RECEIVER MUST NOT mandate the use of the xsi:type attribute in messages except as required in order to indicate a derived type (see XML Schema Part 1: Structures, Section 2.6.1).	y		2. messaging

Rqmt #	Requirement Text	Accepted	Comments / Restrictions	Requirement Type
R1018	The media type of a MESSAGE's envelope MUST indicate the correct character encoding, using the charset parameter.	y		2. messaging
R1025	A RECEIVER MUST handle envelopes in such a way that it appears that all checking of mandatory header blocks is performed before any actual processing.	n	C2C basic profile will not use SOAP headers.	2. messaging
R1027	A RECEIVER MUST generate a "soap:MustUnderstand" fault when an ... not understand.	n	See R1025.	2. messaging
R1028	When a fault is generated by a RECEIVER, further processing SHOULD NOT be performed on the SOAP envelope aside from that which is necessary to rollback, or compensate for, any effects of processing the envelope prior to the generation of the fault.	y		2. messaging
R1029	Where the normal outcome of processing a SOAP envelope would have resulted in the transmission of a SOAP response, but rather a fault is generated instead, a RECEIVER MUST transmit a fault in place of the response	y		2. messaging
R1030	A RECEIVER that generates a fault SHOULD notify the end user that a fault has been generated when practical, by whatever means is deemed appropriate to the circumstance.	y		2. messaging
R1031	When an ENVELOPE contains a faultcode element the content of that element SHOULD NOT use of the SOAP 1.1 "dot" notation to refine the meaning of the fault.	y		2. messaging
R1032	The soap:Envelope, soap:Header, and soap:Body elements in an ENVELOPE MUST NOT have attributes in the namespace "http://schemas.xmlsoap.org/soap/envelope/".	y		2. messaging
R1033	An ENVELOPE SHOULD NOT contain the namespace declaration xmlns:xml="http://www.w3.org/XML/1998/namespace".	y		2. messaging
R1034	A DESCRIPTION SHOULD NOT contain the namespace declaration xmlns:xml="http://www.w3.org/XML/1998/namespace".	y		2. messaging
R1107	A RECEIVER MUST interpret a SOAP message as a Fault when the soap:Body of the message has a single soap:Fault child.	y		2. messaging
R1108	A MESSAGE MUST NOT use the HTTP Extension Framework (RFC2774).	y		2. messaging

Rqmt #	Requirement Text	Accepted	Comments / Restrictions	Requirement Type
R1109	The value of the SOAPAction HTTP header field in a HTTP request MESSAGE MUST be a quoted string.	y		2. messaging
R1110	An INSTANCE MAY accept connections on TCP port 80 (HTTP).	y		2. messaging
R1111	An INSTANCE SHOULD use a "200 OK" HTTP status code on a response message that contains an envelope that is not a fault.	y		2. messaging
R1112	An INSTANCE SHOULD use either a "200 OK" or "202 Accepted" HTTP status code for a response message that does not contain a SOAP envelope but indicates the successful outcome of a HTTP request.	y		2. messaging
R1113	An INSTANCE SHOULD use a "400 Bad Request" HTTP status code, if a HTTP request message is malformed.	y		2. messaging
R1114	An INSTANCE SHOULD use a "405 Method not Allowed" HTTP status code if a HTTP request message's method is not "POST".	y		2. messaging
R1115	An INSTANCE SHOULD use a "415 Unsupported Media Type" HTTP status code if a HTTP request message's Content-Type header field-value is not permitted by its WSDL description.	y		2. messaging
R1119	A RECEIVER MAY respond with a fault if the value of the SOAPAction HTTP header field in a message is not quoted.	y		2. messaging
R1120	An INSTANCE MAY use the HTTP state mechanism ("Cookies").	y	Cookies may be needed once a service consumer has been authenticated.	2. messaging
R1121	An INSTANCE SHOULD NOT require consumer support for Cookies in order to function correctly.	y		2. messaging
R1122	An INSTANCE using Cookies SHOULD conform to RFC2965.	y		2. messaging
R1123	The value of the cookie MUST be considered to be opaque by the CONSUMER.	y		2. messaging
R1124	An INSTANCE MUST use a 2xx HTTP status code on a response message that indicates the successful outcome of a HTTP request.	y		2. messaging
R1125	An INSTANCE MUST use a 4xx HTTP status code for a response that indicates a problem with the format of a request.	y		2. messaging

Rqmt #	Requirement Text	Accepted	Comments / Restrictions	Requirement Type
R1126	An INSTANCE MUST return a "500 Internal Server Error" HTTP status code if the response envelope is a Fault.	y		2. messaging
R1127	A RECEIVER MUST NOT rely on the value of the SOAPAction HTTP header to correctly process the message.	y	C2C basic profile will not use SOAP headers.	2. messaging
R1130	An INSTANCE MUST use the "307 Temporary Redirect" HTTP status code when redirecting a request to a different endpoint.	n		2. messaging
R1131	A CONSUMER MAY automatically redirect a request when it encounters a "307 Temporary Redirect" HTTP status code in a response.	n	See R1130	2. messaging
R1132	A HTTP request MESSAGE MUST use the HTTP POST method.	y		2. messaging
R1140	A MESSAGE SHOULD be sent using HTTP/1.1.	y		2. messaging
R1141	A MESSAGE MUST be sent using either HTTP/1.1 or HTTP/1.0.	y		2. messaging
R2001	A DESCRIPTION MUST only use the WSDL "import" statement to import another WSDL description.	y		3. service description
R2002	To import XML Schema Definitions, a DESCRIPTION MUST use the XML Schema "import" statement.	y		3. service description
R2003	A DESCRIPTION MUST use the XML Schema "import" statement only within the xsd:schema element of the types section.	y		3. service description
R2004	In a DESCRIPTION the schemaLocation attribute of an xsd:import element MUST NOT resolve to any document whose root element is not "schema" from the namespace "http://www.w3.org/2001/XMLSchema".	y		3. service description
R2005	The targetNamespace attribute on the wsdl:definitions element of a description that is being imported MUST have same the value as the namespace attribute on the wsdl:import element in the importing DESCRIPTION.	y		3. service description
R2007	A DESCRIPTION MUST specify a non-empty location attribute on the wsdl:import element.	y		3. service description

Rqmt #	Requirement Text	Accepted	Comments / Restrictions	Requirement Type
R2008	A CONSUMER MAY, but need not, retrieve a WSDL description from the URI specified in the location attribute on a wsdl:import element.	y		3. service description
R2009	An XML Schema directly or indirectly imported by a DESCRIPTION MAY include the Unicode Byte Order Mark (BOM).	y		3. service description
R2010	An XML Schema directly or indirectly imported by a DESCRIPTION MUST use either UTF-8 or UTF-16 encoding.	y	Restricted to UTF-8.	3. service description
R2011	An XML Schema directly or indirectly imported by a DESCRIPTION MUST use version 1.0 of the eXtensible Markup Language W3C Recommendation.	y		3. service description
R2020	The wsdl:documentation element MAY occur as a child of the wsdl:import element in a DESCRIPTION.	y		3. service description
R2021	The wsdl:documentation element MAY occur as a child of the wsdl:part element in a DESCRIPTION.	y		3. service description
R2022	When they appear in a DESCRIPTION, wsdl:import elements MUST precede all other elements from the WSDL namespace except wsdl:documentation.	y		3. service description
R2023	When they appear in a DESCRIPTION, wsdl:types elements MUST precede all other elements from the WSDL namespace except wsdl:documentation and wsdl:import.	y		3. service description
R2024	The wsdl:documentation element MAY occur as a first child of the wsdl:definitions element in a DESCRIPTION.	y		3. service description
R2025	A DESCRIPTION containing WSDL extensions MUST NOT use them to contradict other requirements of the Profile.	y		3. service description
R2026	A DESCRIPTION SHOULD NOT include extension elements with a wsdl:required attribute value of "true" on any WSDL construct (wsdl:binding, wsdl:portType, wsdl:message, wsdl:types, or wsdl:import) that claims conformance to the Profile.	y		3. service description

Rqmt #	Requirement Text	Accepted	Comments / Restrictions	Requirement Type
R2027	If during the processing of a description, a consumer encounters a WSDL extension element that has a wsdl:required attribute with a boolean value of "true" that the consumer does not understand or cannot process, the CONSUMER MUST fail processing.	y		3. service description
R2028	A DESCRIPTION using the WSDL namespace (prefixed "wsdl" in this Profile) MUST be valid according to the XML Schema found at "http://schemas.xmlsoap.org/wsdl/2003-02-11.xsd".	y		3. service description
R2029	A DESCRIPTION using the WSDL SOAP binding namespace (prefixed "soapbind" in this Profile) MUST be valid according to the XML Schema found at "http://schemas.xmlsoap.org/wsdl/soap/2003-02-11.xsd".	y		3. service description
R2030	In a DESCRIPTION the wsdl:documentation element MAY be present as the first child element of wsdl:import, wsdl:part and wsdl:definitions in addition to the elements cited in the WSDL1.1 specification.	y		3. service description
R2101	A DESCRIPTION MUST NOT use QName references to WSDL components in namespaces that have been neither imported, nor defined in the referring WSDL document.	y		3. service description
R2102	A QName reference to a Schema component in a DESCRIPTION MUST use ... on an xsd:import element within the xsd:schema element.	y		3. service description
R2105	All xsd:schema elements contained in a wsdl:types element of a DESCRIPTION MUST have a targetNamespace attribute with a valid and non-null value, UNLESS the xsd:schema element has xsd:import and/or xsd:annotation as its only child element(s).	y		3. service description
R2110	In a DESCRIPTION, declarations MUST NOT extend or restrict the soapenc:Array type.	y		3. service description
R2111	In a DESCRIPTION, declarations MUST NOT use wsdl:arrayType attribute in the type declaration.	y		3. service description
R2112	In a DESCRIPTION, elements SHOULD NOT be named using the convention ArrayOfXXX.	y		3. service description

Rqmt #	Requirement Text	Accepted	Comments / Restrictions	Requirement Type
R2113	An ENVELOPE MUST NOT include the soapenc:arrayType attribute.	y		3. service description
R2114	The target namespace for WSDL definitions and the target namespace for schema definitions in a DESCRIPTION MAY be the same.	y		3. service description
R2201	A document-literal binding in a DESCRIPTION MUST, in each of its soapbind:body element(s), have at most one part listed in the parts attribute, if the parts attribute is specified.	y		3. service description
R2202	A wsdl:binding in a DESCRIPTION MAY contain soapbind:body element(s) that specify that zero parts form the soap:body.	y		3. service description
R2203	An rpc-literal binding in a DESCRIPTION MUST refer, in its soapbind:body element(s), only to wsdl:part element(s) that have been defined using the type attribute.	n	Only document-literal bindings will be used in the c2c basic profile.	3. service description
R2204	A document-literal binding in a DESCRIPTION MUST refer, in each of its soapbind:body element(s), only to wsdl:part element(s) that have been defined using the element attribute.	y		3. service description
R2205	A wsdl:binding in a DESCRIPTION MUST refer, in each of its soapbind:header, soapbind:headerfault and soapbind:fault elements, only to wsdl:part element(s) that have been defined using the element attribute.	y		3. service description
R2206	A wsdl:message in a DESCRIPTION containing a wsdl:part that uses the element attribute MUST refer, in that attribute, to a global element declaration.	y		3. service description
R2207	A wsdl:message in a DESCRIPTION MAY contain wsdl:parts that use the elements attribute provided those wsdl:parts are not referred to by a soapbind:body in an rpc-literal binding.	y		3. service description

Rqmt #	Requirement Text	Accepted	Comments / Restrictions	Requirement Type
R2208	A binding in a DESCRIPTION MAY contain soapbind:header element(s) that refer to wsdl:parts in the same wsdl:message that are referred to by its soapbind:body element(s).	y		3. service description
R2209	A wsdl:binding in a DESCRIPTION SHOULD bind every wsdl:part of a wsdl:message in the wsdl:portType to which it refers with a binding extension element.	y		3. service description
R2210	If a document-literal binding in a DESCRIPTION does not specify the parts attribute on a soapbind:body element, the corresponding abstract wsdl:message MUST define zero or one wsdl:parts.	y		3. service description
R2211	A MESSAGE described with an rpc-literal binding MUST NOT have the xsi:nil attribute with a value of "1" or "true" on the part accessors.	n	See 2203	3. service description
R2212	An ENVELOPE MUST contain exactly one part accessor element for each of the wsdl:part elements bound to the envelope's corresponding soapbind:body element.	y		3. service description
R2213	In a doc-literal description where the value of the parts attribute of soapbind:body is an empty string, the corresponding ENVELOPE MUST have no element content in the soap:Body element.	y		3. service description
R2214	In a rpc-literal description where the value of the parts attribute of soapbind:body is an empty string, the corresponding ENVELOPE MUST have no part accessor elements.	n	C2C basic profile will not use RPC.	3. service description
R2301	The order of the elements in the soap:body of an ENVELOPE MUST be the same as that of the wsdl:parts in the wsdl:message that describes it.	y		3. service description
R2302	A DESCRIPTION MAY use the parameterOrder attribute of an wsdl:operation element to indicate the return value and method signatures as a hint to code generators.	y		3. service description

Rqmt #	Requirement Text	Accepted	Comments / Restrictions	Requirement Type
R2303	A DESCRIPTION MUST NOT use Solicit-Response and Notification type operations in a wsdl:portType definition.	y		3. service description
R2304	A wsdl:portType in a DESCRIPTION MUST have operations with distinct values for their name attributes.	y		3. service description
R2305	A wsdl:operation element child of a wsdl:portType element in a DESCRIPTION MUST be constructed so that the parameterOrder attribute, if present, omits at most 1 wsdl:part from the output message.	y		3. service description
R2306	A wsdl:message in a DESCRIPTION MUST NOT specify both type and element attributes on the same wsdl:part.	y		3. service description
R2401	A wsdl:binding element in a DESCRIPTION MUST use WSDL SOAP Binding as defined in WSDL 1.1 Section 3.	y		3. service description
R2701	The wsdl:binding element in a DESCRIPTION MUST be constructed so that its soapbind:binding child element specifies the transport attribute.	y		3. service description
R2702	A wsdl:binding in a DESCRIPTION MUST specify the HTTP transport protocol with SOAP binding. Specifically, the transport attribute of its soapbind:binding child MUST have the value "http://schemas.xmlsoap.org/soap/http".	y		3. service description
R2705	A wsdl:binding in a DESCRIPTION MUST either be a rpc-literal binding or a document-literal binding.	y	C2C Basic Profile will only use doc literal.	3. service description
R2706	A wsdl:binding in a DESCRIPTION MUST use the value of "literal" for the use attribute in all soapbind:body, soapbind:fault, soapbind:header, and soapbind:headerfault elements.	y		3. service description
R2707	A wsdl:binding in a DESCRIPTION that contains one or more ... MUST be interpreted as though the value "literal" had been specified in each case.	y		3. service description

Rqmt #	Requirement Text	Accepted	Comments / Restrictions	Requirement Type
R2709	A wsdl:portType in a DESCRIPTION MAY have zero or more wsdl:bindings that refer to it, defined in the same or other WSDL documents.	y		3. service description
R2710	The operations in a wsdl:binding in a DESCRIPTION MUST result in operation signatures that are different from one another.	y		3. service description
R2711	A DESCRIPTION SHOULD NOT have more than one wsdl:port with the same value for the location attribute of the soapbind:address element.	y		3. service description
R2712	A document-literal binding MUST be serialized as an ENVELOPE with a soap:Body whose child element is an instance of the global element declaration referenced by the corresponding wsdl:message part.	y		3. service description
R2714	For one-way operations, an INSTANCE MUST NOT return a HTTP response that contains a SOAP envelope. Specifically, the HTTP response entity-body must be empty.	y		3. service description
R2716	A document-literal binding in a DESCRIPTION MUST NOT have the namespace attribute specified on contained soapbind:body, soapbind:header, soapbind:headerfault and soapbind:fault elements.	y		3. service description
R2717	An rpc-literal binding in a DESCRIPTION MUST have the namespace attribute specified, the value of which MUST be an absolute URI, on contained soapbind:body elements.	n	C2C basic profile will not use RPC.	3. service description
R2718	A wsdl:binding in a DESCRIPTION MUST have the same set of wsdl:operations as the wsdl:portType to which it refers.	y		3. service description
R2719	A wsdl:binding in a DESCRIPTION MAY contain no soapbind:headerfault elements if there are no known header faults.	n	No headers.	3. service description
R2720	A wsdl:binding in a DESCRIPTION MUST use the part attribute with a schema type of "NMTOKEN" on all contained soapbind:header and soapbind:headerfault elements.	n	No headers.	3. service description

Rqmt #	Requirement Text	Accepted	Comments / Restrictions	Requirement Type
R2721	A wsdl:binding in a DESCRIPTION MUST have the name attribute specified on all contained soapbind:fault elements.	y		3. service description
R2722	A wsdl:binding in a DESCRIPTION MAY specify the use attribute on contained soapbind:fault elements.	y		3. service description
R2723	If in a wsdl:binding in a DESCRIPTION the use attribute on a contained soapbind:fault element is present, its value MUST be "literal".	y		3. service description
R2724	If an INSTANCE receives an envelope that is inconsistent with its WSDL description, it SHOULD generate a soap:Fault with a faultcode of "Client", unless a "MustUnderstand" or "VersionMismatch" fault is generated.	y		3. service description
R2725	If an INSTANCE receives an envelope that is inconsistent with its WSDL description, it MUST check for "VersionMismatch", "MustUnderstand" and "Client" fault conditions in that order.	y		3. service description
R2726	An rpc-literal binding in a DESCRIPTION MUST NOT have the namespace attribute specified on contained soapbind:header, soapbind:headerfault and soapbind:fault elements.	n	C2C basic profile will not use RPC.	3. service description
R2727	For one-way operations, a CONSUMER MUST NOT interpret a successful HTTP response status code (i.e., 2xx) to mean the message is valid or that the receiver would process it.	y		3. service description
R2728	A wsdl:binding in a DESCRIPTION that omits the use attribute on a contained soapbind:fault element MUST be interpreted as though use="literal" had been specified.	y		3. service description
R2729	An ENVELOPE described with an rpc-literal binding that is a response MUST have a wrapper element whose name is the corresponding wsdl:operation name suffixed with the string "Response".	n	C2C basic profile will not use RPC.	3. service description
R2735	A MESSAGE described with an rpc-literal binding MUST place the part accessor elements for parameters and return value in no namespace.	n	C2C basic profile will not use RPC.	3. service description

Rqmt #	Requirement Text	Accepted	Comments / Restrictions	Requirement Type
R2737	An ENVELOPE described with an rpc-literal binding MUST namespace qualify the descendents of part accessor elements for the parameters and the return value, as defined by the schema in which the part accessor types are defined.	n	C2C basic profile will not use RPC.	3. service description
R2738	An ENVELOPE MUST include all soapbind:headers specified on a wsdl:input or wsdl:output of a wsdl:operation of a wsdl:binding that describes it.	n	No headers.	3. service description
R2739	An ENVELOPE MAY contain SOAP header blocks that are not described in the wsdl:binding that describes it.	n	No headers.	3. service description
R2740	A wsdl:binding in a DESCRIPTION SHOULD contain a soapbind:fault describing each known fault.	y		3. service description
R2741	A wsdl:binding in a DESCRIPTION SHOULD contain a soapbind:headerfault describing each known header fault.	n	No headers.	3. service description
R2742	An ENVELOPE MAY contain fault with a detail element that is not described by a soapbind:fault element in the corresponding WSDL description.	y		3. service description
R2743	An ENVELOPE MAY contain the details of a header processing related fault in a SOAP header block that is not described by a soapbind:headerfault element in the corresponding WSDL description.	n	No headers.	3. service description
R2744	A HTTP request MESSAGE MUST contain a SOAPAction HTTP header field with a quoted value equal to the value of the soapAction attribute of soapbind:operation, if present in the corresponding WSDL description.	y		3. service description
R2745	A HTTP request MESSAGE MUST contain a SOAPAction HTTP header field with a quoted empty string value, if in the corresponding WSDL description, the soapAction of soapbind:operation is either not present, or present with an empty string as its value.	y		3. service description
R2747	A CONSUMER MUST understand and process ... of the value of the wsdl:required attribute, when present.	y		3. service description

Rqmt #	Requirement Text	Accepted	Comments / Restrictions	Requirement Type
R2748	A CONSUMER MUST NOT interpret the presence of the wsdl:required attribute on a soapbind extension element with a value of "false" to mean the extension element is optional in the envelopes generated from the WSDL description.	y		3. service description
R2749	A wsdl:binding in a DESCRIPTION MUST NOT use the parts attribute on contained soapbind:header and soapbind:headerfault elements.	n	No headers.	3. service description
R2750	A CONSUMER MUST ignore an envelope carried in a HTTP response message in a one-way operation.	y	Accepted with MUST changed to SHOULD. If the consumer of a one-way service processes the SOAP response, it only affects performance, not interoperability.	3. service description
R2751	The order of soapbind:header elements in soapbind:binding sections of a DESCRIPTION MUST be considered independent of the order of SOAP header blocks in the envelope.	n	No headers.	3. service description
R2752	A MESSAGE MAY contain more than one instance of each SOAP header block for each soapbind:header element in the appropriate child of soapbind:binding in the corresponding description.	n	No headers.	3. service description
R2753	An ENVELOPE containing SOAP header blocks that are not described in the appropriate wsdl:binding MAY have the mustUnderstand attribute on such SOAP header blocks set to '1'.	n	No headers.	3. service description
R2754	In a DESCRIPTION, the value of the name attribute on a soapbind:fault element MUST match the value of the name attribute on its parent wsdl:fault element.	y		3. service description
R2755	The part accessor elements in a MESSAGE described with an rpc-literal binding MUST have a local name of the same value as the name attribute of the corresponding wsdl:part element.	n	C2C basic profile will not use RPC.	3. service description
R2800	A DESCRIPTION MAY use any construct from XML Schema 1.0.	y		4. schema

Rqmt #	Requirement Text	Accepted	Comments / Restrictions	Requirement Type
R2801	A DESCRIPTION MUST use XML Schema 1.0 Recommendation as the basis of user defined datatypes and structures.	y		4. schema
R2803	In a DESCRIPTION, the namespace attribute of the wsdl:import MUST NOT be a relative URI.	y		4. schema
R3002	REGDATA of type uddi:tModel representing a conformant Web service type MUST use WSDL as the description language.	n	no uddi	5. discovery
R3003	REGDATA of type uddi:tModel representing a conformant Web service type MUST be categorized using the uddi:types taxonomy and a categorization of "wsdlSpec".	n	no uddi	5. discovery
R3010	REGDATA of type uddi:tModel representing a conformant Web service type MUST follow V1.08 of the UDDI Best Practice for Using WSDL in a UDDI Registry.	n	no uddi	5. discovery
R3011	The wsdl:binding that is referenced by REGDATA of type uddi:tModel MUST itself conform to the profile.	n	no uddi	5. discovery
R3020	REGDATA of type uddi:tModel claiming conformance with a Profile MUST be categorized using the ws-i-org:conformsTo:2002_12 taxonomy.	n	See R0005.	1. conformance
R3021	A REGISTRY MUST support the WS-I Conformance category system by adding the ws-i-org:conformsTo:2002_12 tModel definition to its registry content.	n	See R0005.	1. conformance
R3030	REGDATA of type uddi:tModel claiming conformance with a Profile MUST use the ws-i-org:conformsTo:2002_12 categorization value corresponding to the conformance claim URI for that Profile.	n	See R0005.	1. conformance
R3100	REGDATA of type uddi:bindingTemplate representing a conformant INSTANCE MUST contain the uddi:accessPoint element.	n	no uddi	5. discovery
R4001	A RECEIVER MUST accept messages that include the Unicode Byte Order Mark (BOM).	y		2. messaging
R4002	A DESCRIPTION MAY include the Unicode Byte Order Mark (BOM).	y		3. service description
R4003	A DESCRIPTION MUST use either UTF-8 or UTF-16 encoding.	y	Restricted to UTF-8.	3. service description

Rqmt #	Requirement Text	Accepted	Comments / Restrictions	Requirement Type
R4004	A DESCRIPTION MUST use version 1.0 of the eXtensible Markup Language W3C Recommendation.	y		3. service description
R4005	A DESCRIPTION SHOULD NOT contain the namespace declaration xmlns:xml="http://www.w3.org/XML/1998/namespace".	y		3. service description
R5000	An INSTANCE MAY require the use of HTTPS.	y		6. security
R5001	If an INSTANCE requires the use of HTTPS, the location attribute of the soapbind:address element in its wsdl:port description MUST be a URI whose scheme is "https"; otherwise it MUST be a URI whose scheme is "http".	y		6. security
R5010	An INSTANCE MAY require the use of HTTPS with mutual authentication.	y		6. security

§