

NTCIP 2306 v01

National Transportation
Communications for ITS Protocol

Application Profile for XML Message
Encoding and Transport in ITS
Center-to-Center Communications

A Joint Standard of AASHTO, ITE, and NEMA

version 01.69

A Joint Standard of AASHTO, ITE, and NEMA

NTCIP 2306 version v01

National Transportation Communications for ITS Protocol Application Profile for XML Message Encoding and Transport in ITS Center-to-Center Communications

published in December 2008

Published by

American Association of State Highway and Transportation Officials (AASHTO)

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 1752
Rosslyn, Virginia 22209-3801

NOTICES

Copyright Notice

© 2008 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.

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 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 standard are the trademarks of their respective owners.

< This page is intentionally left blank. >

ACKNOWLEDGEMENTS

NTCIP 2306 v01 was prepared by the NTCIP Center-to-Center Working Group (C2C WG), which is a subdivision of the Joint Committee on the NTCIP. The Joint Committee on the NTCIP 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.

When NTCIP 2306 v01 was prepared, the following individuals were members of the NTCIP C2C WG:

- Michael Brown
- Jeff Brummond
- Russ Elovitz
- 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 include:

- Blake Christie
- Tom Passin
- Raman Patel
- Robert Rausch

In addition to the many volunteer efforts, recognition is also given to those organizations that supported the efforts of the NTCIP C2C WG by providing comments and funding, including:

- U.S. Department of Transportation, Research and Innovative Technology Administration
- Caltrans
- ConSysTec
- General Dynamics
- Illinois DOT
- Iteris, Inc.
- Minnesota DOT
- Mitretek
- New York State DOT
- PB Farradyne
- RK Patel Associates
- Siemens ITS
- Southwest Research Institute
- SubCarrier Systems
- TransCore
- Trevilon Corp.

Recognition is also given to liaisons from ITS message set standards developing groups that have supported the efforts of the NTCIP C2C WG so that NTCIP 2306 v01 may be used with the related ITS message set standards. This includes members from the following:

- Advanced Traffic Management Data Dictionary and External Message Sets for Traffic Management Center Communications (an ITE / AASHTO joint standard)
- Transit Communications Interface Profile (a standard of APTA)
- SAE-J2354, Message Sets for Advanced Traveler Information Systems (a standard of SAE)
- IEEE-1512, Standard for Common Incident Management Message Sets for Use by Emergency Management Centers (a standard of IEEE)

FOREWORD

NTCIP 2306 v01, an NTCIP standards publication, defines an application profile for communications between transportation management systems, using Internet standards based on the Extensible Markup Language (XML). NTCIP 2306 v01 defines requirements and optional and conditional clauses that are applicable to specific environments for which they are intended. NTCIP 2306 v01 uses only metric units.

NTCIP 2306 v01 is also an NTCIP Application Profile. An NTCIP Application Profile defines the upper three layers of the ISO seven-layer Open Systems Interconnect (OSI) Reference Model. For more information about NTCIP standards, visit the NTCIP website at 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 standards publication's content. A "Request for Interpretation" of this standards 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 1752
Rosslyn, Virginia 22209-3801
e-mail: ntcip@nema.org

A User Comment should be submitted in the following form:

Standards Publication number and version:
Page:
Section, Paragraph, or Clause:
Comment:
Editorial or Substantive?:
Suggested Alternative Language:

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; March 2008
ITE—Software Standard; March 2008
NEMA—Standard; June 2007

History

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 (FHWA'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 C2C 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 C2C protocol based on web services and the Extensible Markup Language (XML).

The development of NTCIP 2306 v01 started in 2002 under funding provided by FHWA.

NTCIP 2306 v01.51. March 2005—Accepted as a User Comment Draft by the Joint Committee on the NTCIP. March 2005—NTCIP Standards Bulletin B0101 distributed for user comment.

NTCIP 2306 v01.68. March 2006—Accepted as a Recommended Standard by the Joint Committee on the NTCIP. December 2006—NTCIP Standards Bulletin B0115 distributed for ballot and approval.

NTCIP 2306 v01.69. March 2008 to December 2008—Prepared standard for publication; edited format, front matter, and style.

Compatibility of Versions

To distinguish NTCIP 2306 v01 (as published) from previous drafts, NTCIP 2306 v01 also includes NTCIP 2306 v01.69 on each page header. 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.

NTCIP 2306 v01 is designated, and should be cited as, NTCIP 2306 v01. Anyone using NTCIP 2306 v01 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 NTCIP 2306 v01 should seek information about the version number that is of interest to them in any given circumstance.

INTRODUCTION

NTCIP 2306 v01 represents the first of two identified phases for the development of an XML communications standard for C2C communications in the transportation domain (real-time management of public roads and transit systems).

The purpose of NTCIP 2306 v01 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: *Advanced Traffic Management Data Dictionary and External Message Sets for Traffic Management Center Communications* (an ITE/AASHTO joint standard); *Transit Communications Interface Profile* (a standard of APTA); *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 C2C communications efforts. NTCIP 2306 v01 provides projects with 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.

NTCIP 2306 v01 defines requirements that are applicable to all environments using NTCIP XML for C2C communications, and NTCIP 2306 v01 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.

CONTENTS

	Page
SECTION 1 GENERAL.....	1
1.1 Scope.....	1
1.2 References	1
1.2.1 Normative References.....	1
1.2.2 Other References	2
1.2.3 Contact Information	2
1.3 NTCIP 2306 v01 Organization.....	3
1.4 Conformance Clause.....	4
1.5 Glossary of Acronyms.....	4
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
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
4.2.1 SOAP Message Encoding for Request-Response.....	13
4.2.2 SOAP Messages Used in Subscription-Publication	14
SECTION 5 MESSAGE TRANSPORT.....	16
5.1 Message Transport Using HTTP	16
5.1.1 HTTP Headers.....	16
5.1.2 HTTP Headers for SOAP	16
5.1.3 HTTP Secure Sockets (HTTPS).....	16
5.2 Message Transport Using FTP.....	16
5.2.1 FTP	16
SECTION 6 SUB-PROFILES COMMON	17
6.1 WSDL Common.....	17
6.1.1 General Requirements	17
6.1.2 Required WSDL Sections.....	17
6.1.3 Profile Common WSDL Sections.....	17
6.1.4 Sub-Profile Specific WSDL Sections.....	18
6.2 WSDL Definitions Section	18
6.3 WSDL Types and Schema Section	19
6.4 WSDL Message Sections.....	19
6.4.1 Request-Response Message	19
6.4.2 Subscription Message	20
6.4.3 Publication Message	20
6.5 Description of Transport for Secure Socket Service Endpoints	20

SECTION 7 WSDL FOR SUB-PROFILE SOAP OVER HTTP	21
7.1 WSDL for SOAP Request-Response	21
7.1.1 WSDL for SOAP Request-Response PortType Section	21
7.1.2 WSDL for SOAP Request-Response Binding Section	22
7.1.3 WSDL for SOAP Request-Response Service Section	22
7.2 WSDL for SOAP Subscriber Callback Listener (Subscription-Publication)	23
7.2.1 Description of the Subscription-Publication Message Transmission Pattern	23
7.2.2 WSDL for SOAP Subscriber Callback Listener PortType Section	29
7.2.3 WSDL for SOAP Subscriber Callback Listener Binding Section	30
7.2.4 WSDL for SOAP Subscriber Callback Listener Service Section	30
SECTION 8 WSDL FOR SUB-PROFILE XML OVER HTTP	32
8.1 WSDL for XML over HTTP (XML Document Retrieval by File Name)	32
8.1.1 WSDL for XML Over HTTP GET PortType Section	32
8.1.2 WSDL for XML Over HTTP GET Binding Section	32
8.2 WSDL for XML over HTTP (Request-Response)	33
8.2.1 WSDL for XML Over HTTP POST PortType Section	34
8.2.2 WSDL for XML Over HTTP POST Binding Section	34
8.3 WSDL for XML over HTTP Service Section	34
SECTION 9 WSDL FOR SUB-PROFILE XML OVER FTP	36
9.1 WSDL for XML over FTP (XML Document Retrieval by File Name)	36
9.1.1 Extending WSDL for FTP	36
9.1.2 WSDL for XML over FTP PortType Section	36
9.1.3 WSDL for XML over FTP Binding Section	37
SECTION 10 TEST PLAN DEVELOPMENT AND CONFORMANCE GUIDANCE	39
10.1 WSDL Development Methodology	39
10.2 Testing the Correctness of the WSDL	39
ANNEX A SAMPLE WSDL WORKSHEET	40
ANNEX B EXAMPLE—WSDL FOR TMDD	42
B.1 Example—WSDL (Unqualified Names)	42
B.2 Example—WSDL (Qualified Names)	46
ANNEX C EXAMPLE—TMDD MESSAGES	51
C.1 Example—XML-Encoded DMS Inventory Request Message	51
C.2 Example—XML Encoded DMS Inventory Response Message	51
C.3 Example—SOAP-Encoded DMS Inventory Request Message	52
C.4 Example—SOAP-Encoded DMS Inventory Response Message	53
C.5 Example—SOAP-Encoded C2C Message Receipt Message	53
C.6 Example—SOAP-Encoded DMS Inventory Subscription Message	54
C.7 Example—SOAP-Encoded DMS Inventory Publication Message	55

FIGURES

	Page
Figure 1 SOAP Request-Response	21
Figure 2 Subscription	23
Figure 3 Publication	23
Figure 4 XML Document Retrieval using HTTP GET	32
Figure 5 XML Message Exchange with HTTP POST	34
Figure 6 XML Document Retrieval using FTP GET	36

TABLES

	Page
Table 1 Profile Needs.....	6
Table 2 Profile Requirements.....	6
Table 3 Profile Requirements to Solution Trace (Profile Requirements List)	10
Table 4 Sample WSDL Worksheet	41

< This page intentionally left blank. >

Section 1 GENERAL

1.1 SCOPE

NTCIP 2306 v01 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

Normative references contain provisions that, through reference in this text, constitute provisions of NTCIP 2306 v01. Other references in NTCIP 2306 v01 might provide a complete understanding of the entire protocol and the relations between all parts of the protocol. 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 standard listed.

1.2.1 Normative References

W3C	<i>Web Services Description Language (WSDL) 1.1—W3C Note</i> published March 15, 2001
W3C	<i>Simple Object Access Protocol (SOAP) 1.1</i> published May 8, 2000
W3C	<i>Extensible Markup Language (XML) 1.0</i> published February 4, 2004
W3C	<i>Namespaces in XML</i> published January 14, 1999
IETF	<i>Hypertext Transfer Protocol HTTP/1.1 RFC 2616</i> published June 1999
IETF	<i>File Transfer Protocol (FTP) RFC 959</i> published October 1985
IETF	<i>Gzip File Format Specification, version 4.3 RFC 1952</i> published May 1996
IETF	<i>Multipart Internet Mail Extensions (MIME) RFCs 2045, 2046, 2047</i> published November 1996
SAE-J2354	<i>Message Sets for Advanced Traveler Information System (ATIS), Revision 2.0</i> published January 2004

1.2.2 Other References

AASHTO / ITE / NEMA NTCIP 9010 v01	<i>Information Report, XML in Center-to-Center Communications</i> published November 23, 2004
ITE / AASHTO	<i>Traffic Management Center-to-Center Communications {Advanced Traffic Management Data Dictionary (TMDD) and Message Sets (MS)} version 2.1</i> published June 30, 2004
IETF	<i>DEFLATE Compressed Data Format Specification, version 1.3 RFC 1951</i> published May 1996
IEEE-1512	<i>Standard for Common Incident Management Message Sets for Use by Emergency Management Centers</i> published 2006

1.2.3 Contact Information

1.2.3.1 NTCIP Standards

For revision information on this NTCIP standard, contact:

NTCIP Coordinator
National Electrical Manufacturers Association
1300 North 17th Street, Suite 1752
Rosslyn, VA 22209-3801
e-mail: ntcip@nema.org

For draft revisions to this NTCIP standard, and recommended revisions of the NTCIP Joint Committee, visit www.ntcip.org.

1.2.3.2 APTA Standards

APTA standards may be obtained from:

American Public Transportation Association (APTA)
1666 K Street, N.W., Suite 1100
Washington, DC 20006
(202) 496-4800
www.aptastandards.com

1.2.3.3 IEEE Standards

IEEE standards may be obtained from:

IEEE (originally Institute for Electrical and Electronics Engineers, Inc.)
445 Hoes Lane,
Piscataway, NJ 08854-4141
(732) 981-0060
www.ieee.org/portal/site

1.2.3.4 RFC Documents

Request for Comment (RFC) documents may be obtained from several repositories on the World Wide Web, or by “anonymous” File Transfer Protocol (FTP) with several hosts. Browse or FTP to:

www.rfc-editor.org
www.rfc-editor.org/repositories.html
for FTP sites, read <ftp://ftp.isi.edu/in-notes/rfc-retrieval.txt>

1.2.3.5 SAE Standards

SAE standards may be obtained from:

Society of Automotive Engineers (SAE)
755 W. Big Beaver, Suite 1600
Troy MI 48084 USA
(248) 273-2455
www.sae.org/standardsdev/

1.3 NTCIP 2306 V01 ORGANIZATION

NTCIP 2306 v01 consists of ten sections and three annexes.

- a) Section 1—General, provides an overview and introductory material.
- b) Section 2—C2C Messaging Needs and Derived Requirements, 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.
- c) Section 3—NTCIP C2C XML Profile Solutions, contains: a trace of requirements to solutions, and a trace of solutions (sub-profiles) to the document sections that apply (PRL).
- d) Section 4—Message Encoding, provides a normative reference for the encoding of XML and SOAP messages.
- e) Section 5—Message Transport, provides a normative reference for transport of XML and SOAP encoded messages over HTTP and FTP.
- f) Section 6—Sub Profiles Common, provides a normative reference for the description of a web service using the WSDL, message encoding, and transport common across the sub-profiles.
- g) Section 7—WSDL Sub-Profile for SOAP over HTTP, provides a normative reference for the description of a web service using the WSDL for SOAP encoded messages using HTTP. Section 7 supports two transmission patterns: request-response, and publish-subscribe.
- h) Section 8—WSDL Sub-Profile for XML over HTTP, provides a normative reference for the description of a web service using the WSDL for XML encoded messages using HTTP. Section 8 supports the transmission of XML over HTTP with two transmission patterns: request only, and request-response.
- i) Section 9—WSDL Sub-Profile for XML over FTP, provides a normative reference for the description of a web service using WSDL for file transfer of XML encoded messages using FTP. Section 9 supports one transmission pattern: request only.
- j) Section 10—Test Plan Development and Conformance Guidance, provides a brief overview of: WSDL development; how WSDL can be used in the development of more comprehensive test plans for testing center interfaces that use NTCIP 2306 v01, and, therefore, WSDL; and methods for testing the correctness of the syntax of WSDL that conforms with NTCIP 2306 v01.
- k) Annex A—Example WSDL Worksheet for TMDD Dialogs, provides examples of TMDD dialogs and the XML version of the TMDD messages involved in each.
- l) Annex B—Example WSDL for TMDD, contains two examples of WSDL documents based on Annex A. One example shows the use of qualified names and the other unqualified names. (A qualified name contains a prefix indicating the namespace.)
- m) Annex C—Example TMDD Messages, contains example TMDD messages showing proper encoding of messages for XML encoding (appropriate for HTTP and FTP message transport), and SOAP encodings for both SOAP request-response and SOAP subscription-publication message patterns.

1.4 CONFORMANCE CLAUSE

A center system conforms with NTCIP 2306 v01 as follows:

WSDL document(s) SHOULD exist that conform with the normative sections of NTCIP 2306 v01. If WSDL exists, then the XML Schema documents referenced by WSDL SHALL exist.

The WSDL syntax can be shown to be correct per the method outlined in Section 10.

A center system can be shown to support the web services described in the conforming WSDL, including:

- a) XML messages validate against a referenced schema
- b) message inputs and outputs described in the WSDL can be reproduced under a system test
- c) messages can be exchanged using the transports defined in the WSDL

1.5 GLOSSARY OF ACRONYMS

The following glossary defines acronyms used in NTCIP 2306 v01.

ASP	Active Server Pages
ATIS	Advanced Traveler Information Systems
C2C	Center-to-Center
DSRC	Dedicated Short Range Communications
FTP	File Transfer Protocol (an IETF file transfer protocol)
gzip	An IETF file compression standard
HTTP	Hypertext Transfer Protocol
HTTPS	Secure HTTP (Hypertext Transfer Protocol)
IETF	Internet Engineering Task Force
IM	Incident Management
IT IS	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
PERL	Practical Extraction and Report Language
PHP	Hypertext Preprocessor
PN	Profile Need
PR	Profile Requirement
PRL	Profile Requirements List
RFC	Request for Comment
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
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

NTCIP 2306 v01 uses examples based on the concept of operations, dialogs, and messages defined in ITE/AASHTO Traffic Management Center-to-Center Communications (Advanced Traffic Management Data Dictionary (TMDD) and Message Sets (MS)) version 2.1, hereafter referred to as the 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. For example:

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

NTCIP 9010 v01.06 identifies two different XML-based communications mechanisms.

The first mechanism is called a Web Service, enabled by the use of XML-SOAP (Simple Object Access Protocol) and WSDL (Web Services Description Language). SOAP and WSDL are discussed later in this document. Web Services support consumer-initiated requests to a center and the corresponding responses to the consumer. Web Services also support subscription services, where a center can make a one-time request to another center that results in supplier-initiated messages being sent automatically at periodic intervals or whenever data changes. The data frames defined in message sets to support request-response and subscription dialogs fit well with the Web Services paradigm.

The second 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 efficiency 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 straightforward. 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 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 indicate the messaging patterns and encoding needs described in the table below. Sample TMDD message dialogs are described in Annex A. The needs of

other C2C domains and message sets (e.g., ATIS, Incident Management, and Transit) have been found to be similar, and NTCIP 2306 v01 is intended to be applicable within and between all ITS domains.

The need for private connections derives from the use of the internet for C2C communications that may involve transfers of sensitive data and device commands. See Table 1.

Table 1 Profile Needs

Profile Need (PN)	Description
PN 1.0	Message and File Transfer
PN 1.1	This profile needs a mechanism for the transfer of bulk data. This allows an external center to initiate transfer of a file from the information supplier center.
PN 2.0	Message Encoding
PN 2.1	For this profile, messages need to be encoded as XML.
PN 2.2	Privacy. This profile needs a mechanism to allow messages to be privately transmitted over the internet and other shared networks.
PN 3.0	Message Exchange Patterns
PN 3.1	Request-Response. This profile needs a request-response message transmission pattern, described as the communication of a single message sent from an external center (the request) and a single message response from the information supplier center.
PN 3.2	Subscription. This profile needs a message subscription pattern, described as the ability of an external center to subscribe for automatic and ongoing publication of information from an information supplier center. A subscription is initiated by the external center using a single subscription request message. In response to a subscription request, the information supplier center shall indicate acceptance/rejection of the subscription as well as the reason for the request rejection (if rejected). The subsequent publication messages are the subject of the following need.
PN 3.3	Publication. This profile needs a message publication pattern, described as the communication of an asynchronous response message, from an information supplier center that provides or publishes information, to an external center that has previously subscribed for that information. This exchange is initiated by the publisher.

2.3 XML PROFILE REQUIREMENTS

Table 2 describes the XML profile requirements that derive from profile needs.

Table 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	

Profile Requirement (PR)		PN	Requirement Dependencies
	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		
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

Profile Requirement (PR)		PN	Requirement Dependencies
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 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, 3.3	
	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	

Profile Requirement (PR)		PN	Requirement Dependencies
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

NTCIP 2306 v01 is divided into three separate sub-profiles. A sub-profile is an implementation bundle that consists of the following information:

- a) WSDL Definition Requirements—NTCIP 2306 v01 defines the format of the WSDL, and not the WSDL itself. Message set standards develop WSDL for the messages defined in their message set standards.
- b) Message Encoding
- c) Message Transport

WSDL presents a way for a center to describe the basic format of request to their systems. WSDL is written in XML and is a specification.

NTCIP 2306 v01 allows WSDL specifications to be developed that cover the following combinations of message encoding plus transport. Each combination is called a sub-profile.

The sub-profiles of NTCIP 2306 v01 are:

- a) SOAP over HTTP. Using SOAP encoded messages over the hypertext transfer protocol (HTTP), centers will be able to describe and deploy center interfaces that support the request-response and subscription-publication message patterns.
- b) XML over HTTP. Using XML encoded messages over the HTTP, centers will be able to describe and deploy interfaces that support the request-response (via HTTP POST) and request-only message patterns (HTTP GET). HTTP POST is suitable for the exchange of messages (request-response), while the HTTP GET is suitable for the request of an XML document by name.
- c) XML over FTP. Using the file transfer protocol (FTP), centers will be able to describe interfaces that support XML document requests by name.

3.2 TRACEABILITY TO C2C XML PROFILE REQUIREMENTS

Table 3 shows the solutions that the profile shall use to meet each of the profile requirements. Table 3 represents the NTCIP 2306 v01 Profile Requirements List (PRL). The column labeled “NTCIP 2306 v01 Section” points to where additional material can be found in NTCIP 2306 v01. Table 3 facilitates the ability to customize NTCIP 2306 v01 requirements and normative materials 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 Profile Requirements to Solution Trace (Profile Requirements List)

	Profile Requirements List (PRL)	NTCIP 2306 v01 Section	Mandatory / Optional	Profile Requirement	Project Requirement
1.0	SOAP over HTTP				
	a) WSDL Request-Response		O		
	- WSDL General	6.1	M		
	- Definitions	6.2	M	PR 3.1	

Profile Requirements List (PRL)	NTCIP 2306 v01 Section	Mandatory / Optional	Profile Requirement	Project Requirement
- 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 Subscription-Publication		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)	7.2.2	M	PR 4.2.1	
- Binding (Transport)	7.2.3	M	PR 4.2.2a	
- Service (Transport)	7.2.4	M	PR 4.2.2a	
c) Message Encoding		O		
SOAP for Request-Response	4.2.1, 4.2.2.1, 4.2.2.3	M	PR 1.2, 4.2.1a, 4.2.1b	
SOAP for Subscription-Publication	4.2.2.2, 4.2.2.3, 7.2.1	M	PR 1.2, 4.2.1a, 4.2.1b	
d) Message Transport		O		
HTTP	5.1.2	M	PR 2.1a, 4.1.2a, 4.2.2a	
HTTPS	5.1.3, 6.5	O	PR 2.1b	
2.0 XML over HTTP				
a) WSDL Request Only (HTTP GET)		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)	8.1.1	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 (HTTP POST)		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		O		
XML Text	4.1.2	M	PR 1.1a,	

Profile Requirements List (PRL)	NTCIP 2306 v01 Section	Mandatory / Optional	Profile Requirement	Project Requirement
			4.2.2b	
XML gzip	4.1.2	O	PR 1.1 a, 4.2.2b	
e) Message Transport		O		
HTTP	5.1.1, 5.1.2	M	PR 2.1a, 4.2.2b	
HTTPS	5.1.3, 6.5	O	PR 2.1a, 4.2.2b	
3.0 XML over FTP				
a) WSDL Request Only (XML Direct)		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)	9.1.2	M	PR 4.4.1, 3.4	
- Binding (Transport)	9.1.3	M	PR 4.4.2b	
- Service (Transport)	9.1.4	M	PR 4.4.2b	
b) Message Encoding (one or both of the following)		O		
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		O		
FTP	5.2.1	M	PR 2.2a, 4.4.2b	

Section 4 MESSAGE ENCODING

Section 4 provides a normative reference for the encoding of a message as XML or SOAP. Annex C shows example messages encoded as XML, and SOAP encodings for both the SOAP request-response and SOAP subscription-publication message patterns. Section 7.2.1.3 shows the XML Schema that defines the structure of the C2C subscription message header, publication message header, and message receipt.

4.1 XML MESSAGE ENCODING

4.1.1 XML Text Encoding

The following applies to an XML message or file:

Normative

- a) The XML message or file shall be an XML version 1.0 document.
- b) The character set shall be UTF-8.
- c) The XML message should contain the following header:

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

followed by the message content.

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 Section 4.1.1.

4.2 SOAP MESSAGE ENCODING

4.2.1 SOAP Message Encoding for Request-Response

Informative

The <soap:Header> may contain application specific information not covered by NTCIP 2306 v01.

The following represents a correct form of a SOAP request-response 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

- a) The SOAP message shall consist of a <soap:Envelope> tag with two internal tags: a <soap:Header> tag followed by a <soap:Body> tag.
- b) The <soap:Header> tag is provided to ensure that C2C communications software is able to handle a <soap:Header>, even if no header is provided.
- c) 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.

4.2.2 SOAP Messages Used in Subscription-Publication

Informative

The <soap:header> contains application specific information not covered by NTCIP 2306 v01.

4.2.2.1 SOAP Subscription Message

The following represents a correct form of a SOAP subscription message.

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <!-- C2C Subscription Information -->
    <c2cMessageSubscription>
      <!-- C2C Subscription Message Header Content -->

    </c2cMessageSubscription>

    <!--XML message set standard message content, e.g. MS/ETMCC -->
    <dMSStatusRequest>
      <!-- XML Message Content -->
    </dMSStatusRequest>
  </soap:Body>
</soap:Envelope>
```

Normative

- a) The SOAP message shall consist of a <soap:Envelope> tag with two internal tags: a <soap:Header> tag followed by a <soap:Body> tag.
- b) The <soap:Header> tag is provided to ensure that C2C communications software is able to handle a <soap:Header>, even if no header is provided.
- c) The <soap:Body> shall contain two child tags: <c2cMessageSubscription>, and one containing the message set standard XML. 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.

4.2.2.2 SOAP Publication Message

The following represents a correct form of a SOAP publication message.

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <!-- C2C Publication Information -->
    <c2cMessagePublication>
      <!-- C2C Publication Message Header Content -->

    </c2cMessagePublication>

    <!--XML message set standard message content, e.g. MS/ETMCC -->
    <dMSDeviceStatus>
      <!-- XML Message Content -->
    </dMSDeviceStatus>
  </soap:Body>
</soap:Envelope>
```

Normative

- a) The SOAP message shall consist of a <soap:Envelope> tag with two internal tags: a <soap:Header> tag followed by a <soap:Body> tag.
- b) The <soap:Header> tag is provided to ensure that C2C communications software is able to handle a <soap:Header>, even if no header is provided.
- c) The <soap:Body> shall contain two child tags: <c2cMessagePublication>, and one containing the message set standard XML, for example, a TMDD DMS Status Device response message. Use the <iMWrapper> tag to encapsulate IEEE 1512 messages, and <atisMessage> tag for SAE-J2354 messages. XML Messages shall be capable of being validated using the XML Schema(s) referenced in the WSDL.

4.2.2.3 SOAP Subscription and Publication Receipt Message

After a subscription or publication message is transmitted, the receiver shall respond with a receipt message. The following represents a correct form of a SOAP subscription and publication messages.

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <!--Used for both C2C Subscription or Publication Receipt -->
    <c2cMessageReceipt>
      <!-- C2C Message Receipt Content -->

    </c2cMessageReceipt>
  </soap:Body>
</soap:Envelope>
```

Normative

- a) The SOAP message shall consist of a <soap:Envelope> tag with two internal tags: a <soap:Header> tag followed by a <soap:Body> tag.
- b) The <soap:Header> tag is provided to ensure that C2C communications software is able to handle a <soap:Header>, even if no header is provided.
- c) The <soap:Body> shall contain a single child tag: <c2cMessageReceipt>. The c2cMessageReceipt 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

Section 5 and the following sections provide a normative reference for transport of SOAP and XML encoded messages over HTTP.

5.1.1 HTTP Headers

Normative

The HTTP Headers shall conform with IETF HTTP/1.1 RFC 2616.

5.1.2 HTTP Headers for SOAP

Normative

The SOAP message shall conform with the requirements of W3C SOAP 1.1.

5.1.3 HTTP Secure Sockets (HTTPS)

Normative

The use of Secure Sockets message shall conform with the requirements of IETF HTTP/1.1 RFC 2616.

5.2 MESSAGE TRANSPORT USING FTP

Section 5.2 provides a normative reference for transport of XML encoded files using the FTP.

5.2.1 FTP

Informative

NTCIP 2306 v01 supports only the FTP 'GET'.

Normative

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

Section 6 SUB-PROFILES COMMON

6.1 WSDL COMMON

Section 6.1 provides a normative reference for WSDL that applies to all of the sub-profiles, based on the requirements and solutions identified in the previous sections.

In Section 6.1, representative examples given have been taken from the TMDD schema. Normative requirements that 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

- a) Determining Center Services and Operations to Support—It is the prerogative of each data provider to determine which services and operations to support, and, where applicable, to specify different services or versions of a service to support different classes of users. In other words, a data provider may choose to expose different access privileges, interfaces, and 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.
- b) Use of Default XML Namespaces—The use of default XML namespaces and a target namespace is optional. Many XML toolkits do not emit XML that uses default namespaces. It is therefore recognized that the correct use of default namespaces may make it difficult to get XML toolkits to work properly. Therefore, the use of <definitions> and <wsdl:definitions> is allowed, as are <schema> and <xs:schema>, and the use of default namespaces for WSDL elements is optional. The use, or definition, of a target namespace is also optional for the same reasons, namely, XML toolkits may not use or support them.

Normative

- a) The version of WSDL shall conform with W3C WSDL 1.1.
- b) 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 callback listener) shall document the services using WSDL.
- c) A center may provide different WSDL documents for different classes of users.

6.1.2 Required WSDL Sections

Normative

The WSDL definitions shall consists of 5 child sections as follows:

- a) types/schema
- b) message
- c) portType
- d) binding
- e) service

6.1.3 Profile Common WSDL Sections

Informative

The definitions, types/schemas, and message sections are described in Section 6.1.3 and apply across all sub-profiles.

6.1.3.1 Description of the Contents of the Definitions Section

Section 6.2 primarily includes namespace declarations for the entire document.

6.1.3.2 Description of the Contents of the Types/Schema Section

Section 6.3 is used to define the format of a message, and is where one or more 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

Section 6.4 describes the message section found in the WSDL document, which lists the messages that are used by other portions of NTCIP 2306 v01, in essence defining the list of messages supported by a specific implementation.

The WSDL document contains only those messages and dialogs that the specific deployment supports, and provides 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, referred to in NTCIP 2306 v01 as a dialog.

6.1.4.2 Description of the Contents of the Binding Sections

The binding section defines the transport protocol over which messages are transmitted. In NTCIP 2306 v01, these are: 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 a 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: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/"  
  xmlns:c2c="http://schemas.ntcip.org/wsdl/c2c/"  
>
```


Normative

- a) The <definitions> tag shall include a name attribute.
- b) The targetNamespace may be specified, for example, "http://www.tmdd-service". It is not a working URL reference, although it is commonly mistaken for one.
- c) The xmlns:tns namespace may be specified and be the same as the targetNamespace.
- d) The functional area namespace (here xmlns:tmdd) may be specified and be the same as that in the referenced functional area XML Schema which provides the message definitions (here TMDD).
- e) The following additional schema namespaces may be specified:
 - i. xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
 - ii. xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
 - iii. xmlns="http://schemas.xmlsoap.org/wsdl/"
 - iv. xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
 - v. xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
 - vi. xmlns:ftp="http://schemas.ntcip.org/wsdl/ftp/"
 - vii. xmlns:c2c="http://ntcip-c2c-address"

6.3 WSDL TYPES AND SCHEMA SECTION

Informative

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

```
<types>
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:import namespace="http://www.tmdd-address" schemaLocation="tmdd.xsd"/>
    <xs:import namespace="http://www.ntcip-c2c-address" schemaLocation="c2c.xsd"/>
  </xs:schema>
</types>
```

Normative

- a) The type section shall begin with the <types> tag.
- b) The <schema> tag shall immediately follow the <types> tag.
- c) The <import> tag shall follow 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.
- d) Additional namespaces may be included that are project specific.

6.4 WSDL MESSAGE SECTIONS

6.4.1 Request-Response Message

Informative

The following example represents a correct form for specifying a Request-Response message.

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

Normative

- a) The message section shall specify all top-level messages defined in the XML Schema that apply to the project implementation and used a request-response transmission pattern.
- b) The message name shall be defined by the name of the message type, as defined in the schema, with the prefix "MSG_" at the front part of the name.
- c) The name attribute of the message part shall always "message".

- d) The element attribute of the message part name shall use the form "tmdd:" followed by the name of the XML Schema element being imported into the WSDL.

6.4.2 Subscription Message

Informative

The following example represents a correct form for specifying a subscription message.

```
<message name="MSG_DMSInventorySubscription">  
  <part name="c2cMsgAdmin" element="c2c:c2cMessageSubscription"/>  
  <part name="message" element="tmdd:dMSInventoryRequest"/>  
</message>
```

Normative

- a) The message section shall specify all top-level messages defined in the XML Schema that apply to the project implementation and used as a subscription message.
- b) The message name shall be defined by the name of the message type, as defined in the schema, with the prefix "MSG_" at the front part of the name.
- c) The message shall have 2 parts. The name attribute of the first message part shall always be "c2cMsgAdmin", and the element attribute "c2c:c2cMessageSubscription".
- d) The name attribute of the second message part shall be "message" and the element attribute use the form "tmdd:" followed by the name of the XML Schema element being imported into the WSDL.

6.4.3 Publication Message

Informative

The following example represents a correct form for specifying a publication message.

```
<message name="MSG_DMSInventoryPublication">  
  <part name="c2cMsgAdmin" element="c2c:c2cMessagePublication"/>  
  <part name="message" element="tmdd:dMSInventory"/>  
</message>
```

Normative

- a) The message section shall specify all top-level messages defined in the XML Schema that apply to the project implementation and used as a publication message.
- b) The message name shall be defined by the name of the message type, as defined in the schema, with the prefix "MSG_" at the front part of the name.
- c) The message shall have 2 parts. The name attribute of the first message part shall always be "c2cMsgAdmin", and the element attribute "c2c:c2cMessagePublication".
- d) The name attribute of the second message part shall be "message" and the element attribute use the form "tmdd:" followed by the name of the XML Schema element being imported into the WSDL.

6.5 DESCRIPTION OF TRANSPORT FOR SECURE SOCKET SERVICE ENDPOINTS

The following applies to specifying the use of secure sockets in conjunction with the HTTP:

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

Section 7 WSDL FOR SUB-PROFILE SOAP OVER HTTP

7.1 WSDL FOR SOAP REQUEST-RESPONSE

Figure 1 illustrates the request-response pattern as defined by SOAP.

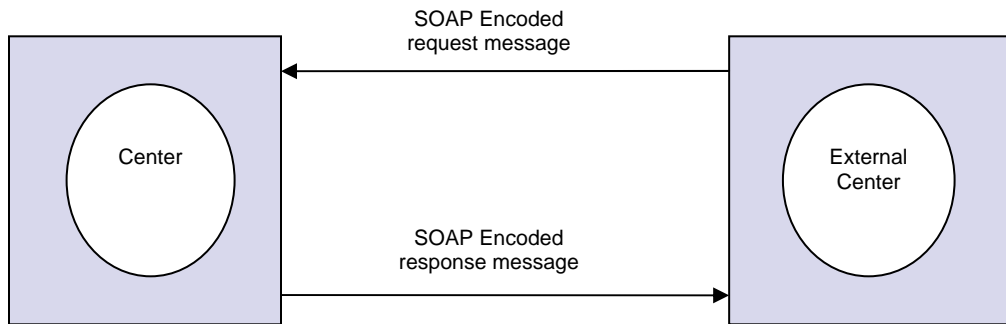


Figure 1 SOAP Request-Response

7.1.1 WSDL for SOAP Request-Response PortType Section

Informative

The following example represents a correct form for specifying a simple request-response portType.

```
<!-- 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

- The <portType> tag shall be the parent tag for a list of operations supported by the SOAP Service for the specific project implementation.
- The operation name shall begin with the prefix "OP_" followed by a descriptive name for the operation.
- Each operation shall have one <input> and one <output> tag. The <input> tag is specified first followed by the <output> tag.
- 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 a correct form for specifying a SOAP over HTTP binding, with 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="requestEventHandler.cgi" style="document"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
```

Normative

- a) The <binding> tag shall be followed by a <soap:binding> tag.
- b) The style attribute of the <soap:binding> tag shall be "document."
- c) The transport attribute of the <soap:binding> tag shall be "http://schemas.xmlsoap.org/soap/http"
- d) The operation name shall match the operation as specified in the portType section of the WSDL for each operation supported by the center.
- e) The <operation> tag shall be followed by the <soap:operation> tag.
- f) The soapAction attribute specifies the soapAction that the external center requester shall include within an HTTP header. There is no requirement for a soapAction, only that the attribute be present. However, if no soapAction is specified, the soapAction attribute shall be written as a double quote followed by two consecutive single quote characters followed by a double quote ("""). The soapAction shall be a URL that indicates the message handler for the endpoint.
- g) The <soap:operation> tag shall be followed by a <input> </input> tag set.
- h) 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.
- i) The <input> tag set shall be followed by an <output> tag set.
- j) 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

WSDL Service sections specify the endpoints that implement the services defined. The following example represents a 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

- a) The <service> tag shall contain a name attribute of the service.
- b) 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.
- c) The <port> tag shall follow the <documentation> tag, if one is present. The name attribute of the port tag reflects the name of a SOAP port followed by the binding attribute indicating the name of the binding identified in the bindings section that applies.
- d) 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.
- e) The service endpoint shall specify a valid URL.
- f) Multiple instances of the <port> element may be defined as needed.

7.2 WSDL FOR SOAP SUBSCRIBER CALLBACK LISTENER (SUBSCRIPTION-PUBLICATION)

Section 7.2 provides for the specification of WSDL for subscribers necessary to implement a listener (data consumer) for publications from a server (data provider).

Figure 2 and Figure 3 show the message exchanges required to support the subscription-publication message transmission pattern. Message exchanges continue until the subscription is cancelled or times out.

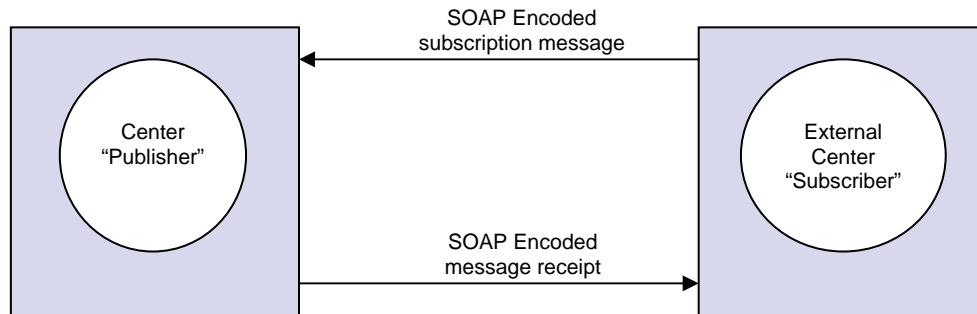


Figure 2 Subscription

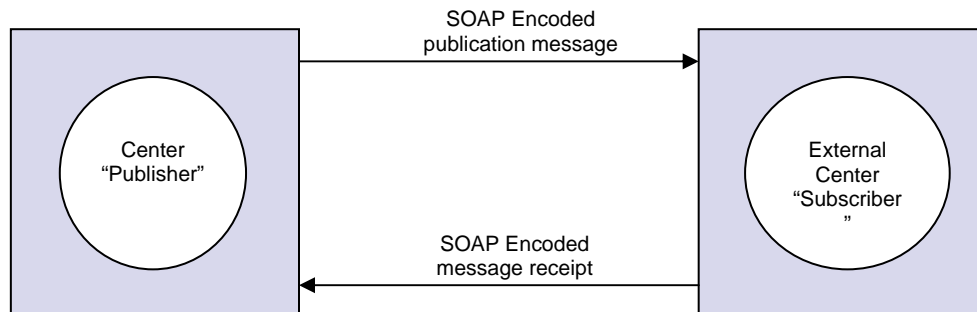


Figure 3 Publication

7.2.1 Description of the Subscription-Publication Message Transmission Pattern

7.2.1.1 Subscription

Normative Responsibilities of the Subscribing Center during Subscription

- a) A subscription shall be initiated by the subscribing center and implemented as a request-response message transmission pattern. The subscribing center initiates a subscription by sending a subscription message to the publishing center. The subscription information shall be included in the subscription header contained in the <soap:Body>, which also contains additional MS-ETMCC, IEEE-

- 1512, or SAE-J2354 XML message information. (See Section 4.1.2 for a description of the proper message encoding rules and Section 7.2.1 for example subscription and publication messages).
- b) The subscribing center shall implement a Subscriber Callback Listener to receive messages during the publication process (see Section 7.2.1.2) issued by the publishing center.
 - c) A subscription message shall include a subscription header to be contained in the <soap:Body> as defined by the schema contained in Section 7.2.1.3. The subscription header contains:
 - i. informationalText (Optional). A text description, 1 to 255 characters, indicating reason for error, if an HTTP error code is returned.
 - ii. returnAddress (Mandatory). This shall be a URL that defines either a SOAP over HTTP endpoint, or an XML over HTTP "POST" endpoint. This allows the publication center to send a message over HTTP.
 - iii. subscriptionType (Mandatory). The subscriptionType may be one of the following:
 - 1) oneTime
 - 2) periodic
 - 3) onChange
 - iv. subscriptionAction (Mandatory). The subscriptionAction includes the following:
 - 1) newSubscription
 - 2) replaceSubscription (restarts a subscription)
 - 3) cancelSubscription
 - 4) cancelAllPriorSubscriptions (subscription ID is ignored)
 - v. subscriptionID (Mandatory). A unique ID within the subscriber's center.
 - vi. subscriptionName (Optional). A human readable text name given by the subscriber's center to identify the subscription.
 - vii. broadcastAlerts (Optional). Value that indicates whether the subscriber grants the publisher use of the subscription-publication channel for broadcast of general alerts not related with the subscription.
 - viii. subscriptionTimeFrame (Optional). The time frame for which the subscription is valid, including:
 - 1) start
 - 2) end
 - ix. subscriptionFrequency (Optional). A number, in seconds, describing the frequency with which the subscriber desires to receive updates.

An example subscription message is shown in Annex C.

Normative Responsibilities of the Publishing Center during Subscription

- a) A publishing center shall receive a subscription message from the subscribing center, and either accept or reject it. Information about any errors shall be contained in the informationalText of the c2cMessageReceipt message, defined in the c2cMessageAdministration XML schema (see Section 7.2.1.3).
- b) The publishing center shall report any HTTP errors to the subscribing center as an HTTP return code.

7.2.1.2 Publication

Normative Responsibilities of the Publishing Center during Publication

- a) The publication of a message shall be implemented as a request-response transmission pattern.
- b) The publishing center initiates the publication of a message to the subscribing center.
- c) The publication center shall specify the WSDL for the subscriber callback listener.
- d) 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).
- e) The publication information shall be included in the "request" portion of the "request-response" message pattern (from publication center to subscriber center).
- f) A publication message shall contain a publication header contained in the <soap:Body> as specified in Section 7.2.1.3. The subscription header contains the following:
 - i. informationalText (Optional). A text description, 1 to 255 characters, indicating reason for error, if an HTTP error code is returned.
 - ii. subscriptionCount (Mandatory). Each published message sent to a subscriber shall include a number starting with 1 through 4,294,967,295 ($2^{32}-1$) and which increments by 1 with each new

message, which identifies the message sequence. This allows 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 4,294,967,295, the count begins with 1 again.

- iii. subscriptionID (Mandatory). 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.
- g) A publishing center may implement the following: upon receiving acknowledgement that a message has been missed, a subscriber may request that the missing message be resent.

An example publication message is shown in Annex C.

Normative Responsibilities of the Subscribing Center during Publication

- a) The subscribing center shall acknowledge the receipt of a publication message, and either accept or reject it. Information about any errors shall be contained in the informationalText of the c2cMessageReceipt message, defined in the c2cMessageAdministration XML schema (see Section 7.2.1.3).
- b) The subscribing center shall report any HTTP errors to the publishing center as an HTTP return code.

7.2.1.3 Subscription-Publication Message Administration XML Schema

The C2C Message Administration XML schema is shown below:

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

<!-- NTCIP 2306 -->
<xs:schema targetNamespace="http://www.ntcip-c2c-address"
  xmlns="http://www.ntcip-c2c-address"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:atis="http://www.dummy-atis-address"

  elementFormDefault="unqualified"
  attributeFormDefault="unqualified"
  version="DRAFT">

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

<xs:annotation>
  <xs:documentation>
    Last Updated: Friday, January 6, 2006
  </xs:documentation>
</xs:annotation>

<!-- Complex Types -->

<!-- Descriptive Name: c2cMessageSubscription -->
<xs:element name="c2cMessageSubscription" type="C2CMessageSubscription"/>
<xs:complexType name="C2CMessageSubscription" >
  <xs:sequence>
    <xs:element name="informationalText" type="InformationalText" minOccurs="0"/>
    <xs:element name="returnAddress" type="ReturnAddress"/>
    <xs:element name="subscriptionAction" type="SubscriptionAction"/>
    <xs:element name="subscriptionType" type="SubscriptionType"/>
    <xs:element name="subscriptionID" type="SubscriptionID"/>
    <xs:element name="subscriptionName" type="SubscriptionName" minOccurs="0"/>
    <xs:element name="subscriptionTimeFrame" type="SubscriptionTimeFrame" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

```
    <xs:element name="subscriptionFrequency" type="SubscriptionFrequency" minOccurs="0"/>
    <xs:element name="broadcastAlerts" type="BroadcastAlerts" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

```
<!-- Descriptive Name: c2cMessagePublication -->
<xs:element name="c2cMessagePublication" type="C2CMessagePublication"/>
<xs:complexType name="C2CMessagePublication" >
  <xs:sequence>
    <xs:element name="informationalText" type="InformationalText" minOccurs="0"/>
    <xs:element name="subscriptionID" type="SubscriptionID"/>
    <xs:element name="subscriptionName" type="SubscriptionName" minOccurs="0"/>
    <xs:element name="subscriptionCount" type="SubscriptionFrequency" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

```
<!-- Descriptive Name: c2cMessageReceipt -->
<xs:element name="c2cMessageReceipt" type="C2CMessageReceipt"/>
<xs:complexType name="C2CMessageReceipt" >
  <xs:sequence>
    <xs:element name="informationalText" type="InformationalText"/>
  </xs:sequence>
</xs:complexType>
```

```
<!-- Descriptive Name: SubscriptionTimeFrame -->
<xs:complexType name="SubscriptionTimeFrame" >
  <xs:annotation>
    <xs:documentation>
      Time frame during which subscriber requests that publication be active.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="start" type="atis:DateTimePair" minOccurs="0"/>
    <xs:element name="end" type="atis:DateTimePair" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

```
<!-- Simple Types -->
```

```
<!-- Descriptive Name: BroadcastAlertsItem -->
<xs:simpleType name="BroadcastAlertsItem" >
  <xs:annotation>
    <xs:appinfo>
      broadcastAlertsAccepted (1)
      broadcastAlertsNotAccepted (2)
    </xs:appinfo>
  </xs:annotation>
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:int">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="2"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:restriction base="xs:string">
```



```
        <xs:enumeration value="broadcastAlertsAccepted"/>
        <xs:enumeration value="broadcastAlertsNotAccepted"/>
    </xs:restriction>
</xs:simpleType >
</xs:union>
</xs:simpleType>
<xs:simpleType name="BroadcastAlerts">
    <xs:list itemType="BroadcastAlertsItem"/>
</xs:simpleType>

<!-- Descriptive Name: ReturnAddress -->
<xs:simpleType name="ReturnAddress" >
    <xs:annotation>
        <xs:documentation>
            A URL indicating the subscriber callback message handler. Any set of ASCII characters up to 128.
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="128"/>
    </xs:restriction>
</xs:simpleType>

<!-- Descriptive Name: SubscriptionActionItem -->
<xs:simpleType name="SubscriptionActionItem" >
    <xs:annotation>
        <xs:appinfo>
            newSubscription (1)
            replaceSubscription (2)
            cancelSubscription (3)
            cancelAllPriorSubscriptions (4)
        </xs:appinfo>
    </xs:annotation>
    <xs:union>
        <xs:simpleType>
            <xs:restriction base="xs:int">
                <xs:minInclusive value="1"/>
                <xs:maxInclusive value="4"/>
            </xs:restriction>
        </xs:simpleType>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="newSubscription"/>
                <xs:enumeration value="replaceSubscription"/>
                <xs:enumeration value="cancelSubscription"/>
                <xs:enumeration value="cancelAllPriorSubscriptions"/>
            </xs:restriction>
        </xs:simpleType >
    </xs:union>
</xs:simpleType>
<xs:simpleType name="SubscriptionAction">
    <xs:list itemType="SubscriptionActionItem"/>
</xs:simpleType>
```

```
<!-- Descriptive Name: SubscriptionCount -->
<xs:simpleType name="SubscriptionCount" >
  <xs:annotation>
    <xs:documentation>
      The nth time the publisher has sent content as part of a description to the subscriber.
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:unsignedInt">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="4294967295"/>
  </xs:restriction>
</xs:simpleType>
```

```
<!-- Descriptive Name: SubscriptionFrequency -->
<xs:simpleType name="SubscriptionFrequency" >
  <xs:annotation>
    <xs:documentation>
      seconds
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:unsignedInt">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="4294967295"/>
  </xs:restriction>
</xs:simpleType>
```

```
<!-- Descriptive Name: SubscriptionID -->
<xs:simpleType name="SubscriptionID" >
  <xs:annotation>
    <xs:documentation>
      ID created by the subscription subscriber. Any set of alphanumeric characters up to 32
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="32"/>
  </xs:restriction>
</xs:simpleType>
```

```
<!-- Descriptive Name: SubscriptionName -->
<xs:simpleType name="SubscriptionName" >
  <xs:annotation>
    <xs:documentation>
      Name created by the subscriber for a subscription. Any set of ASCII characters up to 128
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="128"/>
  </xs:restriction>
</xs:simpleType>
```

```

<!-- Descriptive Name: InformationalText -->
<xs:simpleType name="InformationalText" >
  <xs:annotation>
    <xs:documentation>
      Any set of ASCII characters up to 255
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="255"/>
  </xs:restriction>
</xs:simpleType>

<!-- Descriptive Name: SubscriptionTypeItem -->
<xs:simpleType name="SubscriptionTypeItem" >
  <xs:annotation>
    <xs:appinfo>
      oneTime (1)
      periodic (2)
      onChange (3)
    </xs:appinfo>
  </xs:annotation>
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:int">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="3"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="oneTime"/>
        <xs:enumeration value="periodic"/>
        <xs:enumeration value="onChange"/>
      </xs:restriction>
    </xs:simpleType >
  </xs:union>
</xs:simpleType>
<xs:simpleType name="SubscriptionType">
  <xs:list itemType="SubscriptionTypeItem"/>
</xs:simpleType>

</xs:schema>

```

7.2.2 WSDL for SOAP Subscriber Callback Listener PortType Section

Informative

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

```

<portType name="tmddServiceSubscriberCallbackSOAPPort">
  <operation name="OP_SubscriberEventUpdateInformation">
    <input message="tns:MSG_BasicEventUpdate"/>
    <output message="tns:MSG_C2CMessageReceipt"/>
  </operation>
</portType>

```

Normative

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

7.2.3 WSDL for SOAP Subscriber Callback Listener Binding Section

Informative

The following represents a correct form for specifying a SOAP binding for a subscriber callback listener. This is consistent with the concept that a subscriber (external center acting as a data consumer) provides 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=" " 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 supports -->
</binding>
```

Normative

The normative rules from the SOAP Request-Response Binding defined in Section 7.1.3 shall apply with the following exception—the "soapAction" attribute shall be left blank, to be specified during subscription.

7.2.4 WSDL for SOAP Subscriber Callback Listener Service Section

Informative

The following example represents a correct form for specifying a SOAP Subscriber Callback Service. The SOAP Subscriber Callback Service specifies a template that an external center uses, 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

- a) The <service> tag shall contain a name attribute of the service.
- b) 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.
- c) The <port> tag shall follow the <documentation> tag, if one is present. The name attribute of the port tag reflects the name of a SOAP port followed by the binding attribute indicating the name of the binding identified in the bindings section that applies.
- d) Following the <port> tag, a <soap:address> tag shall be placed. The location attribute of the <soap:address> tag shall specify a fake (placeholder) service endpoint of the SOAP service, which contains the subscriber callback listener. However, to support valid WSDL, the <soap:address> tag shall be included. The subscriber shall provide an endpoint at run-time as described in Section 7.1.2 covering Subscription.

Section 8 WSDL FOR SUB-PROFILE XML OVER HTTP

8.1 WSDL FOR XML OVER HTTP (XML DOCUMENT RETRIEVAL BY FILE NAME)

Figure 4 shows the mechanism for XML document retrieval by file name using HTTP GET.

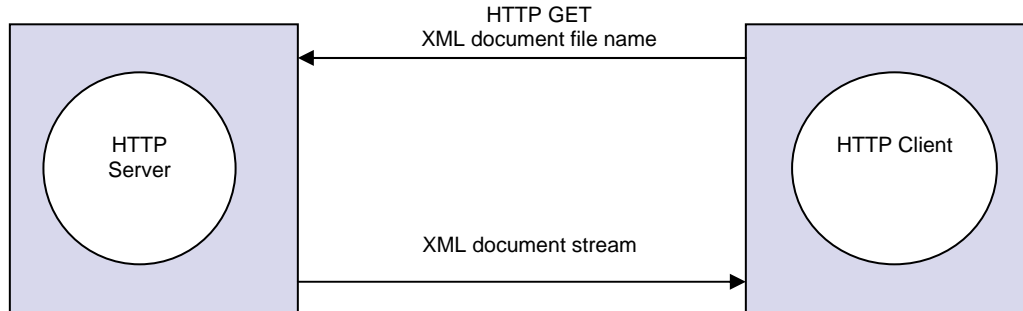


Figure 4 XML Document Retrieval using HTTP GET

8.1.1 WSDL for XML Over HTTP GET PortType Section

Informative

The following represents a 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

- The WSDL to describe the simple retrieval of a file shall be described as a one-way operation.
- The <portType> tag shall be followed by a single operation that maps to a specific file.
- The operation shall be a one-way operation between the data provider center and the external center (data consumer).
- The operation name shall begin with the prefix “OP_” followed by a descriptive name for the operation.
- Each operation shall have one <input> tag.
- 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 GET 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 XML HTTP Service (see Section 9.1.3 g)) with the file name specified 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 a correct form for specifying the binding of operations (as listed in the portType section) for HTTP “GET”. The example represents a correct form for specifying the binding for an XML document over HTTP. The output may be any MIME type, 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

- a) The <binding> tag shall be followed by a <http:binding> tag.
- b) The http:binding verb attribute shall be “GET”.
- c) The operation name shall match the operation as specified in the portType section of the WSDL for each operation (file) supported by the center.
- d) The <operation> tag shall be followed by the <http:operation> tag.
- e) 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.
- f) The <http:operation> tag shall be followed by the <input> </input> tag set.
- g) The <input> tag shall be followed by a <http:urlEncoded> tag.
- h) The <input> tag set shall be followed by an <output> </output> tag set.
- i) The <output> tag shall be followed by a <mime:content> tag.
- j) The valid type attribute of the <mime:content> tag shall be any MIME type.
- k) If the MIME type attribute is “gzip”, then the uncompressed information shall be of type “text/xml”.

8.2 WSDL FOR XML OVER HTTP (REQUEST-RESPONSE)

Section 8.2 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 allows implementation based on CGI and web server scripting such as PERL, ASP, PHP, etc.

Figure 5 shows the mechanism for XML encoded message exchanges using HTTP POST.

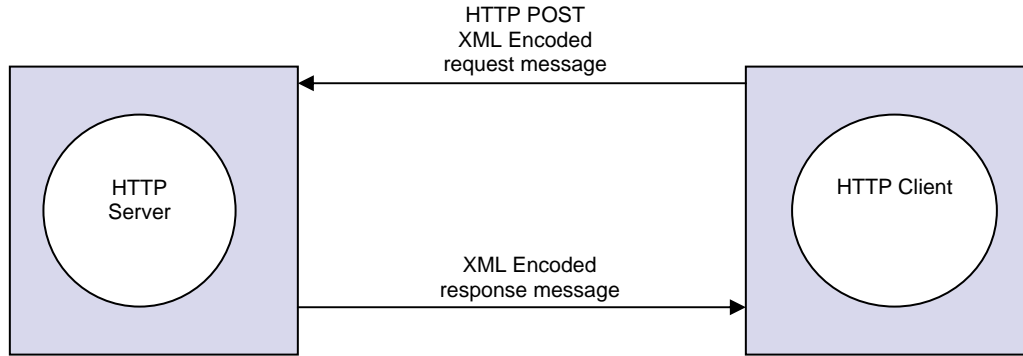


Figure 5 XML Message Exchange with HTTP POST

8.2.1 WSDL for XML Over HTTP POST PortType Section

The portType section for XML over HTTP follows the same rules as those in Section 7.1.1.

8.2.2 WSDL for XML Over HTTP POST Binding Section

Informative

The following example represents a correct form for specifying the binding of operations (as listed in the portType section) for HTTP “POST”. The example represents a 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

- The same requirement of the Request Only Binding shall apply.
- The content of the <input> tag shall contain at least one form parameter, which shall contain an XML message.
- The POST arguments shall be defined by the part attribute information contained in the <message> section of the WSDL. Currently, the part attribute is standardized to be called “message”, which shall be the name of the POST argument contained in a fully developed XML request message. See Section 6.4.1.

8.3 WSDL FOR XML OVER HTTP SERVICE SECTION

Informative

The following represents a 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:

- a) The XML HTTP endpoint uses the <http:address> tag to specify an endpoint rather than the <soap:address> tag.
- b) Also, 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 (XML DOCUMENT RETRIEVAL BY FILE NAME)

Section 9, for use of XML over FTP, is modeled after Section 8, WSDL for XML over HTTP (Request Only File Retrieval), which uses the HTTP "GET". Figure 6 shows the mechanism for XML document retrieval by file name using FTP GET.

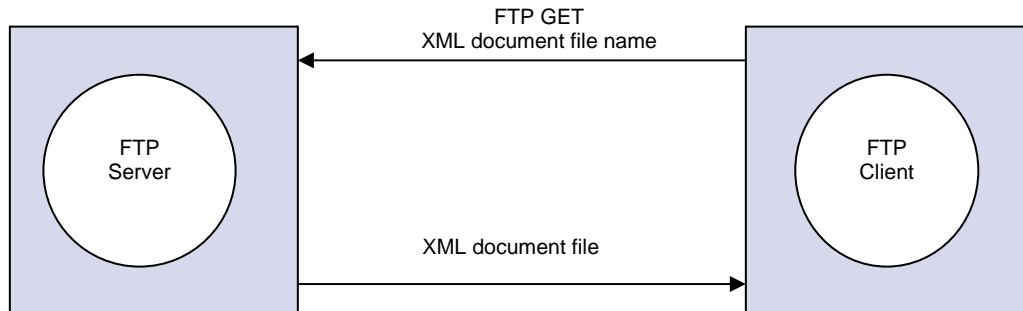


Figure 6 XML Document Retrieval using FTP GET

9.1.1 Extending WSDL for FTP

At the time of this writing, WSDL does not specify an FTP binding. However, since both HTTP and FTP support a 'GET' operation, NTCIP 2306 v01 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 a 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. (Note that neither the operation nor message below is defined within TMDD.)

The following represents a 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:

- The operation name and output message follow the same rules as those defined for a SOAP Service portType.
- 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 a 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:

- a) 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 Section g)) with the file name specified 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 a correct form for specifying the binding of operations (as listed in the portType 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:

- a) The operation name shall match the operation as specified in the portType section of the WSDL for each operation (file) supported by the center.
- b) The <operation> tag shall be followed by the <ftp:operation> tag.
- c) 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.
- d) The <ftp:operation> tag shall be followed by the <input> </input> tag set.
- e) The <input> tag shall be followed by a <ftp:urlEncoded> tag.
- f) The <input> tag set shall be followed by an <output> </output> tag set.
- g) 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 shall be WSDL for XML over FTP Service Section.

The following represents a 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 exceptions:

- a) The XML FTP endpoint uses the <ftp:address> tag to specify an endpoint rather than the <soap:address> tag.
- b) Insert comment about documentation again here.
- c) Also, 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).
- d) 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 MS/ETMCC, is included in Annex A and Annex B.

WSDL is developed solely to describe a web service, so determining which standards-based messages are implemented in the form of a web service by an agency or private company is a necessary step.

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

- a) The initial step was to document the operation, input messages, and output messages of each dialog in table form. (See Annex A.)
- b) From a reading of the documentation included in the TMDD and by interpreting the associated dialog, the dialog was characterized as either: Request-Response, Subscription-Publication, or One-Way.
- c) Finally, a judgment was made as to which dialogs would be reasonable to support using the XMLDirect (XML over HTTP and XML over FTP) one-way methods.
- d) The contents of steps a) to c) are documented in tabular form for each dialog contained in the TMDD dialogs, and shown in Annex A. Annex A was then used as a basis for development of the TMDD Profile needs listed in Section 2.
- e) Finally, a simple straightforward translation of the table shown in Annex A 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.

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 WSDL syntax is to validate the WSDL against the WSDL schema (i.e., the schema that specifies what WSDL should look like).

Annex A

SAMPLE WSDL WORKSHEET

Table 4 is a sample WSDL worksheet that provides examples of TMDD dialogs and the XML version of the TMDD messages involved in each.

Table 4 Sample WSDL Worksheet

Service	Operation	Message Input	Message Output	Message Pattern	Message Encoding	Message Transport
tmddSOAPService	OP_ManageDMSControl	MSG_DMSControlRequest	MSG_DMSControlResponse	R/R	SOAP	HTTP
tmddSOAPService	OP_ManageDMSInventoryRequest	MSG_DMSInventoryRequest	MSG_DMSInventoryResponse	R/R	SOAP	HTTP
tmddSOAPService	OP_ManageDMSInventorySubscription	MSG_DMSInventorySubscription	MSG_C2CMessageReceipt	Sub	SOAP	HTTP
tmddSOAPService	OP_ManageDMSStatusRequest	MSG_DMSStatusRequest	MSG_DMSStatusResponse	R/R	SOAP	HTTP
tmddSOAPService	OP_ManageDMSStatusSubscription	MSG_DMSStatusSubscription	MSG_C2CMessageReceipt	Sub	SOAP	HTTP
tmddSOAPSubscriberCallback	OP_SubscriberDMSInventoryInformation	MSG_C2CMessageReceipt	MSG_DMSInventoryPublication	Pub	SOAP	HTTP
tmddSOAPSubscriberCallback	OP_SubscriberDMSStatusInformation	MSG_C2CMessageReceipt	MSG_DMSStatusPublication	Pub	SOAP	HTTP
tmddHTTPPostService	OP_ManageDMSInventoryRequest	MSG_DMSInventoryRequest	MSG_DMSInventoryResponse	R/R	XML	HTTP POST
tmddHTTPPostService	OP_ManageDMSStatusRequest	MSG_DMSStatusRequest	MSG_DMSStatusResponse	R/R	XML	HTTP POST
tmddXMLDirectService	OP_PublishDMSInventoryInformation		MSG_DMSInventory	Oneway	XML	HTTP GET
tmddXMLDirectService	OP_PublishDMSInventoryInformation		MSG_DMSInventory	Oneway	XML	FTP GET
tmddXMLDirectService	OP_PublishDMSStatusInformation		MSG_DMSDeviceStatus	Oneway	XML	HTTP GET
tmddXMLDirectService	OP_PublishDMSStatusInformation		MSG_DMSDeviceStatus	Oneway	XML	FTP GET

Legend:
R/R = Request-Response
Sub = Subscription
Pub = Publication
Oneway = One Way

Annex B EXAMPLE—WSDL FOR TMDD

Annex B contains two examples of WSDL documents based on Annex A. One example shows the use of unqualified names and the other qualified names. (A qualified name contains a prefix indicating the namespace.)

B.1 EXAMPLE—WSDL (UNQUALIFIED NAMES)

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions
  name="ntcip2306sample"
  targetNamespace="http://www.ntcip.org/ntcip2306sample.wsdl"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:tns="http://www.ntcip.org/ntcip2306sample.wsdl"
  xmlns:c2c="http://www.ntcip-c2c-address"
  xmlns:ftp="http://schemas.ntcip.org/wsdl/ftp/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tmdd="http://www.tmdd-address"
>

  <types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:import namespace="http://www.tmdd-address"
        schemaLocation="TMDD.xsd"/>
      <xsd:import namespace="http://www.ntcip-c2c-address"
        schemaLocation="c2c.xsd"/>
    </xsd:schema>
  </types>

  <message name="MSG_DMSInventoryRequest">
    <part element="tmdd:dMSInventoryRequest" name="message"/>
  </message>
  <message name="MSG_DMSStatusResponse">
    <part element="tmdd:dMSDeviceStatus" name="message"/>
  </message>
  <message name="MSG_DMSStatusSubscription">
    <part element="c2c:c2cMessageSubscription" name="c2msgadmin"/>
    <part element="tmdd:dMSStatusRequest" name="message"/>
  </message>
  <message name="MSG_DMSInventoryResponse">
    <part name="message" element="tmdd:dMSInventory"/>
  </message>
  <message name="MSG_DMSInventorySubscription">
    <part element="c2c:c2cMessageSubscription" name="c2msgadmin"/>
    <part element="tmdd:dMSInventoryRequest" name="message"/>
  </message>
  <message name="MSG_DMSStatusRequest">
    <part element="tmdd:dMSStatusRequest" name="message"/>
  </message>
  <message name="MSG_DMSControlRequest">
    <part element="tmdd:dMSControlRequest" name="message"/>
  </message>
  <message name="MSG_DMSInventoryPublication">
    <part element="c2c:c2cMessagePublication" name="c2msgadmin"/>
    <part element="tmdd:dMSInventory" name="message"/>
  </message>
  <message name="MSG_DMSControlResponse">
    <part element="tmdd:dMSControlResponse" name="message"/>
  </message>
  <message name="MSG_C2CMessageReceipt">
    <part element="c2c:c2cMessageReceipt" name="message"/>
  </message>

```



```

<message name="MSG_DMSStatusPublication">
  <part element="c2c:c2cMessagePublication" name="c2msgadmin"/>
  <part element="tmdd:dMSDeviceStatus" name="message"/>
</message>

<portType name="tmddServiceSubscriberCallbackSOAPPortType">
  <operation name="OP_SubscriberDMSInventoryInformation">
    <input message="tns:MSG_DMSInventoryPublication"/>
    <output message="tns:MSG_C2CMessageReceipt"/>
  </operation>
  <operation name="OP_SubscriberDMSStatusInformation">
    <input message="tns:MSG_DMSStatusPublication"/>
    <output message="tns:MSG_C2CMessageReceipt"/>
  </operation>
</portType>
<portType name="tmddXMLDirectPortType">
  <operation name="OP_PublishDMSInventoryInformation">
    <input message="tns:MSG_DMSInventoryResponse"/>
  </operation>
  <operation name="OP_PublishDMSStatusInformation">
    <input message="tns:MSG_DMSStatusResponse"/>
  </operation>
</portType>
<portType name="tmddHTTPPostPortType">
  <operation name="OP_ManageDMSInventoryRequest">
    <input message="tns:MSG_DMSInventoryRequest"/>
    <output message="tns:MSG_DMSInventoryResponse"/>
  </operation>
  <operation name="OP_ManageDMSStatusRequest">
    <input message="tns:MSG_DMSStatusRequest"/>
    <output message="tns:MSG_DMSStatusResponse"/>
  </operation>
</portType>
<portType name="tmddSOAPPortType">
  <operation name="OP_ManageDMSControl">
    <input message="tns:MSG_DMSSControlRequest"/>
    <output message="tns:MSG_DMSSControlResponse"/>
  </operation>
  <operation name="OP_ManageDMSStatusRequest">
    <input message="tns:MSG_DMSStatusRequest"/>
    <output message="tns:MSG_DMSStatusResponse"/>
  </operation>
  <operation name="OP_ManageDMSStatusSubscription">
    <input message="tns:MSG_DMSStatusSubscription"/>
    <output message="tns:MSG_C2CMessageReceipt"/>
  </operation>
  <operation name="OP_ManageDMSInventoryRequest">
    <input message="tns:MSG_DMSInventoryRequest"/>
    <output message="tns:MSG_DMSInventoryResponse"/>
  </operation>
  <operation name="OP_ManageDMSInventorySubscription">
    <input message="tns:MSG_DMSInventorySubscription"/>
    <output message="tns:MSG_C2CMessageReceipt"/>
  </operation>
</portType>

<binding name="tmddSOAPBinding" type="tns:tmddSOAPPortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="OP_ManageDMSControl">
    <soap:operation soapAction="dmsControl.cgi" style="document"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  <operation name="OP_ManageDMSInventoryRequest">
    <soap:operation soapAction="dmsInventoryRequestHandler.cgi" style="document"/>
    <input>

```

```
        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
</operation>
<operation name="OP_ManageDMSInventorySubscription">
    <soap:operation
        soapAction="dmsInventorySubscriptionHandler.cgi"
        style="document"/>
    <input>
        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
</operation>
<operation name="OP_ManageDMSStatusRequest">
    <soap:operation soapAction="dmsStatusRequestHandler.cgi" style="document"/>
    <input>
        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
</operation>
<operation name="OP_ManageDMSStatusSubscription">
    <soap:operation soapAction="dmsStatusSubscriptionHandler.cgi" style="document"/>
    <input>
        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
</operation>
</binding>
<binding name="tmddFTPGetBinding" type="tns:tmddXMLDirectPortType">
    <ftp:binding verb="GET"/>
    <operation name="OP_PublishDMSStatusInformation">
        <http:operation location="dmsStatusInformation.xml"/>
        <input>
            <ftp:urlEncoded/>
        </input>
        <output>
            <mime:content type="text/xml"/>
        </output>
    </operation>
    <operation name="OP_PublishDMSInventoryInformation">
        <http:operation location="dmsInventoryInformation.xml"/>
        <input>
            <ftp:urlEncoded/>
        </input>
        <output>
            <mime:content type="text/xml"/>
        </output>
    </operation>
</binding>
<binding name="tmddHTTPPostBinding" type="tns:tmddHTTPPostPortType">
    <http:binding verb="POST"/>
    <operation name="OP_ManageDMSInventoryRequest">
        <http:operation location="dmsInventoryRequestHandler.cgi"/>
        <input>
            <mime:content type="application/x-www-form-urlencoded"/>
        </input>
        <output>
            <mime:content type="text/xml"/>
        </output>
    </operation>
    <operation name="OP_ManageDMSStatusRequest">
        <http:operation location="dmsStatusRequestHandler.cgi"/>
        <input>
```

```

        <mime:content type="application/x-www-form-urlencoded"/>
    </input>
    <output>
        <mime:content type="text/xml"/>
    </output>
</operation>
</binding>
<binding name="tmddHTTPGetBinding" type="tns:tmddXMLDirectPortType">
    <http:binding verb="GET"/>
    <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>
<binding
    name="tmddServiceSubscriberCallbackSOAPBinding"
    type="tns:tmddServiceSubscriberCallbackSOAPPortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="OP_SubscriberDMSInventoryInformation">
        <soap:operation soapAction="" style="document"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
    <operation name="OP_SubscriberDMSStatusInformation">
        <soap:operation soapAction="" style="document"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
</binding>
<service name="tmddXMLDirectService">
    <documentation>Traffic Management XMLDirect Service</documentation>
    <port binding="tns:tmddFTPGetBinding" name="tmddFTPGetPort">
        <ftp:address location="ftp://www.mycenter.org/c2cxml/tmdd"/>
    </port>
    <port binding="tns:tmddHTTPGetBinding" name="tmddHTTPGetPort">
        <http:address location="http://www.mycenter.org/c2cxml/tmdd"/>
    </port>
</service>
<service name="tmddSOAPService">
    <documentation>Traffic Management Service</documentation>
    <port binding="tns:tmddSOAPBinding" name="tmddSOAPPort">
        <soap:address location="http://www.mycenter.org/c2cxml/tmdd/tmddSOAPService"/>
    </port>
</service>
<service name="tmddHTTPPostService">
    <documentation>Traffic Management HTTP POST Service</documentation>
    <port binding="tns:tmddHTTPPostBinding" name="tmddHTTPPostPort">
        <http:address location="http://www.mycenter.org/c2cxml/tmdd"/>
    </port>
</service>

```

```
        </port>
    </service>
    <service name="tmddServiceSubscriberCallback">
        <documentation>Subscribers for my center's Traffic Management Service</documentation>
        <port
            binding="tns:tmddServiceSubscriberCallbackSOAPBinding"
            name="tmddServiceSubscriberCallbackSOAPPort">
            <soap:address location="http://www.tmddservice-subscriber-callback"/>
        </port>
    </service>
</definitions>
```

B.2 EXAMPLE—WSDL (QUALIFIED NAMES)

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
    name="ntcip2306sample"
    targetNamespace="http://www.ntcip.org/ntcip2306sample.wsdl"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:tns="http://www.ntcip.org/ntcip2306sample.wsdl"
    xmlns:c2c="http://www.ntcip-c2c-address"
    xmlns:ftp="http://schemas.ntcip.org/wsdl/ftp/"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tmdd="http://www.tmdd-address"
    >

    <wsdl:types>
        <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
            <xsd:import namespace="http://www.tmdd-address"
                schemaLocation="TMDD.xsd"/>
            <xsd:import namespace="http://www.ntcip-c2c-address"
                schemaLocation="c2c.xsd"/>
        </xsd:schema>
    </wsdl:types>

    <wsdl:message name="MSG_DMSInventoryRequest">
        <wsdl:part element="tmdd:dMSInventoryRequest" name="message"/>
    </wsdl:message>
    <wsdl:message name="MSG_DMSStatusResponse">
        <wsdl:part element="tmdd:dMSDeviceStatus" name="message"/>
    </wsdl:message>
    <wsdl:message name="MSG_DMSStatusSubscription">
        <wsdl:part element="c2c:c2cMessageSubscription" name="c2msgadmin"/>
        <wsdl:part element="tmdd:dMSStatusRequest" name="message"/>
    </wsdl:message>
    <wsdl:message name="MSG_DMSInventoryResponse">
        <wsdl:part name="message" element="tmdd:dMSInventory"/>
    </wsdl:message>
    <wsdl:message name="MSG_DMSInventorySubscription">
        <wsdl:part element="c2c:c2cMessageSubscription" name="c2msgadmin"/>
        <wsdl:part element="tmdd:dMSInventoryRequest" name="message"/>
    </wsdl:message>
    <wsdl:message name="MSG_DMSStatusRequest">
        <wsdl:part element="tmdd:dMSStatusRequest" name="message"/>
    </wsdl:message>
    <wsdl:message name="MSG_DMSControlRequest">
        <wsdl:part element="tmdd:dMSControlRequest" name="message"/>
    </wsdl:message>
    <wsdl:message name="MSG_DMSInventoryPublication">
        <wsdl:part element="c2c:c2cMessagePublication" name="c2msgadmin"/>
        <wsdl:part element="tmdd:dMSInventory" name="message"/>
    </wsdl:message>
    <wsdl:message name="MSG_DMSControlResponse">
        <wsdl:part element="tmdd:dMSControlResponse" name="message"/>
    </wsdl:message>
    <wsdl:message name="MSG_C2CMessageReceipt">
        <wsdl:part element="c2c:c2cMessageReceipt" name="message"/>
    </wsdl:message>
    <wsdl:message name="MSG_DMSStatusPublication">
```

```

        <wsdl:part element="c2c:c2cMessagePublication" name="c2msgadmin" />
        <wsdl:part element="tmdd:dMSDeviceStatus" name="message" />
    </wsdl:message>

    <wsdl:portType name="tmddServiceSubscriberCallbackSOAPPortType">
        <wsdl:operation name="OP_SubscriberDMSInventoryInformation">
            <wsdl:input message="tns:MSG_DMSInventoryPublication" />
            <wsdl:output message="tns:MSG_C2CMessageReceipt" />
        </wsdl:operation>
        <wsdl:operation name="OP_SubscriberDMSStatusInformation">
            <wsdl:input message="tns:MSG_DMSStatusPublication" />
            <wsdl:output message="tns:MSG_C2CMessageReceipt" />
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:portType name="tmddXMLDirectPortType">
        <wsdl:operation name="OP_PublishDMSInventoryInformation">
            <wsdl:input message="tns:MSG_DMSInventoryResponse" />
        </wsdl:operation>
        <wsdl:operation name="OP_PublishDMSStatusInformation">
            <wsdl:input message="tns:MSG_DMSStatusResponse" />
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:portType name="tmddHTTPPostPortType">
        <wsdl:operation name="OP_ManageDMSInventoryRequest">
            <wsdl:input message="tns:MSG_DMSInventoryRequest" />
            <wsdl:output message="tns:MSG_DMSInventoryResponse" />
        </wsdl:operation>
        <wsdl:operation name="OP_ManageDMSStatusRequest">
            <wsdl:input message="tns:MSG_DMSStatusRequest" />
            <wsdl:output message="tns:MSG_DMSStatusResponse" />
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:portType name="tmddSOAPPortType">
        <wsdl:operation name="OP_ManageDMSControl">
            <wsdl:input message="tns:MSG_DMSControlRequest" />
            <wsdl:output message="tns:MSG_DMSControlResponse" />
        </wsdl:operation>
        <wsdl:operation name="OP_ManageDMSStatusRequest">
            <wsdl:input message="tns:MSG_DMSStatusRequest" />
            <wsdl:output message="tns:MSG_DMSStatusResponse" />
        </wsdl:operation>
        <wsdl:operation name="OP_ManageDMSStatusSubscription">
            <wsdl:input message="tns:MSG_DMSStatusSubscription" />
            <wsdl:output message="tns:MSG_C2CMessageReceipt" />
        </wsdl:operation>
        <wsdl:operation name="OP_ManageDMSInventoryRequest">
            <wsdl:input message="tns:MSG_DMSInventoryRequest" />
            <wsdl:output message="tns:MSG_DMSInventoryResponse" />
        </wsdl:operation>
        <wsdl:operation name="OP_ManageDMSInventorySubscription">
            <wsdl:input message="tns:MSG_DMSInventorySubscription" />
            <wsdl:output message="tns:MSG_C2CMessageReceipt" />
        </wsdl:operation>
    </wsdl:portType>

    <wsdl:binding name="tmddSOAPBinding" type="tns:tmddSOAPPortType">
        <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
        <wsdl:operation name="OP_ManageDMSControl">
            <soap:operation soapAction="dmsControl.cgi" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="OP_ManageDMSInventoryRequest">
            <soap:operation soapAction="dmsInventoryRequestHandler.cgi" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
        </wsdl:operation>
    </wsdl:binding>

```

```
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="OP_ManageDMSInventorySubscription">
  <soap:operation
    soapAction="dmsInventorySubscriptionHandler.cgi"
    style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="OP_ManageDMSStatusRequest">
  <soap:operation soapAction="dmsStatusRequestHandler.cgi" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="OP_ManageDMSStatusSubscription">
  <soap:operation soapAction="dmsStatusSubscriptionHandler.cgi" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="tmddFTPGetBinding" type="tns:tmddXMLDirectPortType">
  <ftp:binding verb="GET"/>
  <wsdl:operation name="OP_PublishDMSStatusInformation">
    <http:operation location="dmsStatusInformation.xml"/>
    <wsdl:input>
      <ftp:urlEncoded/>
    </wsdl:input>
    <wsdl:output>
      <mime:content type="text/xml"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="OP_PublishDMSInventoryInformation">
    <http:operation location="dmsInventoryInformation.xml"/>
    <wsdl:input>
      <ftp:urlEncoded/>
    </wsdl:input>
    <wsdl:output>
      <mime:content type="text/xml"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="tmddHTTPPostBinding" type="tns:tmddHTTPPostPortType">
  <http:binding verb="POST"/>
  <wsdl:operation name="OP_ManageDMSInventoryRequest">
    <http:operation location="dmsInventoryRequestHandler.cgi"/>
    <wsdl:input>
      <mime:content type="application/x-www-form-urlencoded"/>
    </wsdl:input>
    <wsdl:output>
      <mime:content type="text/xml"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="OP_ManageDMSStatusRequest">
    <http:operation location="dmsStatusRequestHandler.cgi"/>
    <wsdl:input>
      <mime:content type="application/x-www-form-urlencoded"/>
```

```

        </wsdl:input>
        <wsdl:output>
            <mime:content type="text/xml"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="tmddHTTPGetBinding" type="tns:tmddXMLDirectPortType">
    <http:binding verb="GET"/>
    <wsdl:operation name="OP_PublishDMSInventoryInformation">
        <http:operation location="dmsInventoryInformation.xml"/>
        <wsdl:input>
            <http:urlEncoded/>
        </wsdl:input>
        <wsdl:output>
            <mime:content type="text/xml"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="OP_PublishDMSStatusInformation">
        <http:operation location="dmsStatusInformation.xml"/>
        <wsdl:input>
            <http:urlEncoded/>
        </wsdl:input>
        <wsdl:output>
            <mime:content type="text/xml"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:binding
    name="tmddServiceSubscriberCallbackSOAPBinding"
    type="tns:tmddServiceSubscriberCallbackSOAPPortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="OP_SubscriberDMSInventoryInformation">
        <soap:operation soapAction="" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="OP_SubscriberDMSStatusInformation">
        <soap:operation soapAction="" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>

<wsdl:service name="tmddXMLDirectService">
    <wsdl:documentation>Traffic Management XMLDirect Service</wsdl:documentation>
    <wsdl:port binding="tns:tmddFTPGetBinding" name="tmddFTPGetPort">
        <ftp:address location="ftp://www.mycenter.org/c2cxml/tmdd/">
    </wsdl:port>
    <wsdl:port binding="tns:tmddHTTPGetBinding" name="tmddHTTPGetPort">
        <http:address location="http://www.mycenter.org/c2cxml/tmdd/">
    </wsdl:port>
</wsdl:service>
<wsdl:service name="tmddSOAPService">
    <wsdl:documentation>Traffic Management Service</wsdl:documentation>
    <wsdl:port binding="tns:tmddSOAPBinding" name="tmddSOAPPort">
        <soap:address location="http://www.mycenter.org/c2cxml/tmdd/tmddSOAPService/">
    </wsdl:port>
</wsdl:service>
<wsdl:service name="tmddHTTPPostService">
    <wsdl:documentation>Traffic Management HTTP POST Service</wsdl:documentation>
    <wsdl:port binding="tns:tmddHTTPPostBinding" name="tmddHTTPPostPort">
        <http:address location="http://www.mycenter.org/c2cxml/tmdd/">
    </wsdl:port>

```

```
</wsdl:service>
<wsdl:service name="tmddServiceSubscriberCallback">
  <wsdl:documentation>Subscribers for my center's Traffic Management
Service</wsdl:documentation>
  <wsdl:port
    binding="tns:tmddServiceSubscriberCallbackSOAPBinding"
    name="tmddServiceSubscriberCallbackSOAPPort">
    <soap:address location="http://www.tmddservice-subscriber-callback"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```


Annex C EXAMPLE—TMDD MESSAGES

Annex C contains example TMDD messages showing proper encoding of messages for XML encoding (appropriate for HTTP and FTP message transport), and SOAP encodings for both SOAP request-response and SOAP subscription-publication message patterns.

C.1 EXAMPLE—XML ENCODED DMS INVENTORY REQUEST MESSAGE

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

<!-- ***** -->
<!-- Example XML Encoded DMS Inventory Request Message -->
<!-- Only the top level tmdd namespace is defined. Additional namespaces -->
<!-- may need to be declared for elements defined within other schemas, -->
<!-- e.g. atis, lrms -->

<tmdd:dMSInventoryRequest xmlns:tmdd="http://www.tmdd-address">
  <inventory-request>
    <organization-owning>
      <organization-id>
        tmc.dot.state.org
      </organization-id>
      <organization-name>
        The State Department of Transportation
      </organization-name>
    </organization-owning>

    <organization-requesting>
      <organization-id>
        tmc.pwd.city.state.org
      </organization-id>
      <organization-name>
        The City Traffic Division
      </organization-name>
    </organization-requesting>

    <device-list>
      <device>
        <device-id>
          <!-- request all -->
          0
        </device-id>
      </device>
    </device-list>

  </inventory-request>
</tmdd:dMSInventoryRequest>
```

C.2 EXAMPLE—XML ENCODED DMS INVENTORY RESPONSE MESSAGE

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

<!-- ***** -->
<!-- Example XML Encoded DMS Inventory Response Message -->
<!-- Only the top level tmdd namespace is defined. Additional namespaces -->
<!-- may need to be declared for elements defined within other schemas, -->
<!-- e.g. atis, lrms -->

<tmdd:dMSInventory xmlns:tmdd="http://www.tmdd-address">
  <device>
    <organization-information>
      <organization-id>
        tmc.dot.state.org
      </organization-id>
    </organization-information>
  </device>
</tmdd:dMSInventory>
```

```
        </organization-id>
        <organization-name>
            The State Department of Transportation
        </organization-name>
    </organization-information>
    <operator-id>
        0
    </operator-id>
    <device-id>
        22
    </device-id>
    <device-name>
        22.dms.dot.state.org
    </device-name>
    <dms-sign-type>
        variable message sign
    </dms-sign-type>
    <device-location>
        <latitude>
            43000000
        </latitude>
        <longitude>
            -74000000
        </longitude>
    </device-location>
    </device>
    <!-- Continue with additional devices -->
</tmdd:dMSInventory>
```

C.3 EXAMPLE—SOAP-ENCODED DMS INVENTORY REQUEST MESSAGE

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

<!-- ***** -->
<!-- Example SOAP Encoded DMS Inventory Request Message -->
<!-- Only the top level tmdd namespace is defined. Additional namespaces -->
<!-- may need to be declared for elements defined within other schemas, -->
<!-- e.g. atis, lrms -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope">
    <soap:Header>
    </soap:Header>
    <soap:Body>
        <tmdd:dMSInventoryRequest xmlns:tmdd="http://www.tmdd-address">
            <inventory-request>
                <organization-owning>
                    <organization-id>
                        tmc.dot.state.org
                    </organization-id>
                    <organization-name>
                        The State Department of Transportation
                    </organization-name>
                </organization-owning>

                <organization-requesting>
                    <organization-id>
                        tmc.pwd.city.state.org
                    </organization-id>
                    <organization-name>
                        The City Traffic Division of Public Works Department
                    </organization-name>
                </organization-requesting>

                <device-list>
                    <device>
                        <device-id>
                            <!-- request all -->
                            0
                        </device-id>
                    </device>
                </device-list>
            </inventory-request>
        </tmdd:dMSInventoryRequest>
    </soap:Body>
</soap:Envelope>
```

```

        </device-id>
      </device>
    </device-list>
  </inventory-request>
</tmdd:dMSInventoryRequest>
</soap:Body>
</soap:Envelope>

```

C.4 EXAMPLE—SOAP-ENCODED DMS INVENTORY RESPONSE MESSAGE

```

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

<!-- ***** -->
<!-- Example SOAP Encoded DMS Inventory Response Message -->
<!-- Only the top level tmdd namespace is defined. Additional namespaces -->
<!-- may need to be declared for elements defined within other schemas, -->
<!-- e.g. atis, lrms -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope">
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <tmdd:dMSInventory xmlns:tmdd="http://www.tmdd-address">
      <device>
        <organization-information>
          <organization-id>
            tmc.dot.state.org
          </organization-id>
          <organization-name>
            The State Department of Transportation
          </organization-name>
        </organization-information>
        <operator-id>
          0
        </operator-id>
        <device-id>
          22
        </device-id>
        <device-name>
          22.dms.dot.state.org
        </device-name>
        <dms-sign-type>
          variable message sign
        </dms-sign-type>
        <device-location>
          <latitude>
            43000000
          </latitude>
          <longitude>
            -74000000
          </longitude>
        </device-location>
      </device>
      <!-- Continue with additional devices -->
    </tmdd:dMSInventory>
  </soap:Body>
</soap:Envelope>

```

C.5 EXAMPLE—SOAP-ENCODED C2C MESSAGE RECEIPT MESSAGE

```

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

<!-- ***** -->
<!-- Example SOAP Encoded C2C Message Receipt -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope">
  <soap:Header>
  </soap:Header>

```

```

    <soap:Body>
      <c2c:c2cMessageReceipt xmlns:c2c='http://www.ntcip-c2c-address'>
        <informationalText>
          Example Informational Text.
        </informationalText>
      </c2c:c2cMessageReceipt>
    </soap:Body>
  </soap:Envelope>

```

C.6 EXAMPLE—SOAP-ENCODED DMS INVENTORY SUBSCRIPTION MESSAGE

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- ***** -->
<!-- Example SOAP Encoded DMS Inventory Subscription Message -->
<!-- Only the top level tmdd namespace is defined. Additional namespaces -->
<!-- may need to be declared for elements defined within other schemas, -->
<!-- e.g. atis, lrms -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope">
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <c2c:c2cMessageSubscription xmlns:c2c='http://www.ntcip-c2c-address'>
      <informationalText>
      </informationalText>
      <subscriptionType>
        <!-- reserved (0) -->
        <!-- oneTime (1) -->
        <!-- periodic (2) -->
        <!-- onChange (3) -->
        3
      </subscriptionType>
      <subscriptionID>
        111
      </subscriptionID>
      <subscriptionName>
        State DOT DMS Inventory
      </subscriptionName>
      <returnAddress>
        http://www.subscriberscenter.org/c2cxml/subscriberCallbackMessageHandler.cgi
      </returnAddress>
      <subscriptionAction>
        <!-- reserved (0) -->
        <!-- newSubscription (1) -->
        <!-- replaceSubscription (2) -->
        <!-- cancelSubscription (3) -->
        <!-- cancelAllPriorSubscriptions (4) -->
        1
      </subscriptionAction>
    </c2c:c2cMessageSubscription>
    <tmdd:dMSInventoryRequest xmlns:tmdd="http://www.tmdd-address">
      <inventory-request>
        <organization-owning>
          <organization-id>
            tmc.dot.state.org
          </organization-id>
          <organization-name>
            The State Department of Transportation
          </organization-name>
        </organization-owning>
        <organization-requesting>
          <organization-id>
            tmc.pwd.city.state.org
          </organization-id>
          <organization-name>
            The City Traffic Division of Public Works Department
          </organization-name>
        </organization-requesting>
      </inventory-request>
    </tmdd:dMSInventoryRequest>
  </soap:Body>
</soap:Envelope>

```

```

        <device-list>
          <device>
            <device-id>
              <!-- request all -->
              0
            </device-id>
          </device>
        </device-list>
      </inventory-request>
    </tmdd:dMSInventoryRequest>
  </soap:Body>
</soap:Envelope>

```

C.7 EXAMPLE—SOAP-ENCODED DMS INVENTORY PUBLICATION MESSAGE

```

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

<!-- ***** -->
<!-- Example SOAP Encoded DMS Inventory Publication Message -->
<!-- Only the top level tmdd namespace is defined. Additional namespaces -->
<!-- may need to be declared for elements defined within other schemas, -->
<!-- e.g. atis, lrms -->

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope">
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <c2c:c2cMessagePublication xmlns:c2c='http://www.ntcip-c2c-address'>
      <informationalText>
      </informationalText>
      <!-- ID and Name below are given by the subscriber -->
      <subscriptionID>
        111
      </subscriptionID>
      <subscriptionName>
        State DOT DMS Inventory
      </subscriptionName>
      <subscriptionCount>
        3
      </subscriptionCount>
    </c2c:c2cMessagePublication>

    <tmdd:dMSInventory xmlns:tmdd="http://www.tmdd-address">
      <device>
        <organization-information>
          <organization-id>
            tmc.dot.state.org
          </organization-id>
          <organization-name>
            The State Department of Transportation
          </organization-name>
        </organization-information>
        <operator-id>
          0
        </operator-id>
        <device-id>
          22
        </device-id>
        <device-name>
          22.dms.dot.state.org
        </device-name>
        <dms-sign-type>
          variable message sign
        </dms-sign-type>
      </device>
    </tmdd:dMSInventory>
  </soap:Body>
</soap:Envelope>

```

```
        <device-location>
          <latitude>
            43000000
          </latitude>
          <longitude>
            -74000000
          </longitude>
        </device-location>
      </device>
    <!-- Continue with additional devices -->
  </tmdd:dMSInventory>
</soap:Body>
</soap:Envelope>
```

§