International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.830.2
(07/2016)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

E-health multimedia services and applications – Interoperability compliance testing of personal health systems (HRN, PAN, LAN, TAN and WAN)

Conformance of ITU-T H.810 personal health devices: WAN interface Part 2: Web services interoperability: Receiver

Recommendation ITU-T H.830.2

Recommendation ITU-T H.830.2

Conformance of ITU-T H.810 personal health devices: WAN interface Part 2: Web services interoperability: Receiver

Summary

Recommendation ITU-T H.830.2 is a transposition of Continua Health Alliance Test Tool DG2013, Test Suite Structure & Test Purposes, WAN Interface; Part 2: Web Services Interoperability. Receiver (Version 1.3, 2014-01-24), that was developed by the Continua Health Alliance. A number of versions of this specification existed before transposition.

This Recommendation includes an electronic attachment with the protocol implementation conformance statements (PICS) and the protocol implementation extra information for testing (PIXIT) required for the implementation of Annex A.

This Recommendation was initially approved as ITU-T H.832 (01/2015) and later renumbered, without further modifications, as ITU-T H.830.2 (01/2015) for consistency with the numbering of new WAN interface conformance testing specifications.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T H.832	2015-01-13	16	11.1002/1000/12250
1.0	ITU-T H.830.2	2015-01-13	16	11.1002/1000/12588
2.0	ITU-T H.830.2	2016-07-14	16	11.1002/1000/12922

Keywords

Conformance testing, continua design guidelines, e-health, H.810, WAN interface, personal connected health devices, wide area network.

---

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11 830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Electronic attachment: This Recommendation includes an electronic attachment with the protocol implementation conformance statements (PICS) and the protocol implementation extra information for testing (PIXIT) required for the implementation of Annex A.

Introduction

This Recommendation is a transposition of Continua Health Alliance Test Tool DG2013, Test Suite Structure & Test Purposes, WAN Interface; Part 2: Web Services Interoperability. Receiver (Version 1.3, 2014-01-24), that was developed by the Continua Health Alliance. A number of versions of this specification existed before transposition and these can be found in the table below.

Version	Date	Revision history
1.1	2012-10-05	Initial release for Test Tool DG2011. It is the same version as "TSS&TP_1.5_WAN_PART_2_(REC WS-I)_v1.1.doc" because new features included in [b-CDG 2011] do not affect the test procedures specified in this document.
1.2	2013-05-24	Initial release for Test Tool DG2012. It is the same version as "TSS&TP_DG2011_WAN_PART_2_(REC WS-I)_v1.1.doc" because new features included in [b-CDG 2012] do not affect the test procedures specified in this document.
1.3	2014-01-24	Initial release for Test Tool DG2013. It is the same version as "TSS&TP_DG2012_WAN_PART_2_(REC WS-I)_v1.1.doc" because new features included in CDG 2013 [b-ITU-T H.810 (2013)] do not affect the test procedures specified in this document.

Recommendation ITU-T H.830.2

Conformance of ITU-T H.810 personal health devices: WAN interface Part 2: Web services interoperability: Receiver

1   Scope

The scope of this Recommendation[1] is to provide a test suite structure and the test purposes (TSS & TP) for the WAN interface based on the requirements defined in the Continua Design Guidelines (CDG) [ITU-T H.810 (2015)]. The objective of this test specification is to provide a high probability of air interface interoperability between different devices.

TSS & TP for the WAN interface have been divided into the eight parts specified below. This Recommendation covers Part 2.

– **Part 1:** Web services interoperability [ITU-T H.810 (2015)] Sender

– **Part 2: Web services interoperability** [ITU-T H.810 (2015)] **Receiver**

– **Part 3:** SOAP/ATNA. Sender

– **Part 4:** SOAP/ATNA. Receiver

– **Part 5:** PCD-01 HL7 messages. Sender

– **Part 6:** PCD-01 HL7 messages. Receiver

– **Part 7:** Consent management. Sender

– **Part 8:** Consent management. Receiver

2   References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T H.810 (2015)]	Recommendation ITU-T H.810 (2015), *Interoperability design guidelines for personal health systems*.
[ITU-T H.810 (2016)]	Recommendation ITU-T H.810 (2016), *Interoperability design guidelines for personal health systems*.
[IEEE 11073-20601A]	IEEE 11073-20601A-2010, *IEEE Health informatics – Personal health device communication – Part 20601: Application profile – Optimized Exchange Protocol Amendment 1*. <http://standards.ieee.org/findstds/standard/11073-20601a-2010.html>
[OASIS/WS-I BP]	OASIS/WS-I (2006), *Basic Profile Version 1.1.* http://www.ws-i.org/Profiles/BasicProfile-1.1.html

---

[1] This Recommendation includes an electronic attachment with the protocol implementation conformance statements (PICS) and the protocol implementation extra information for testing (PIXIT) required for the implementation of Annex A.

[OASIS/WS-I BSP]   OASIS/WS-I (2007), *WS-I Basic Security Profile Version 1.0.*
http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html

[OASIS WS-I RM]   OASIS (2007), *Reliable Messaging Version 1.1.*
http://docs.oasis-open.org/ws-rx/wsrm/200702/wsrm-1.1-spec-cs-01.pdf

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 agent [IEEE 11073-20601A]: A node that collects and transmits personal health data to an associated manager.

3.1.2 manager [IEEE 11073-20601A]: A node receiving data from one or more agent systems. Some examples of managers include a cellular phone, health appliance, set top box, or a computer system.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AHD	Application Hosting Device
ATNA	Audit Trail and Node Authentication
ATS	Abstract Test Suite
CDG	Continua Design Guidelines
DUT	Device Under Test
GUI	Graphical User Interface
HL7	Health Level 7
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
INR	International Normalized Ratio
IUT	Implementation Under Test
MDS	Medical Device System
NFC	Near Field Communication
PCO	Point of Control and Observation
PCT	Protocol Conformance Testing
PHD	Personal Healthcare Device
PHDC	Personal Healthcare Device Class
PHM	Personal Healthcare Monitoring (report)
PICS	Protocol Implementation Conformance Statement
PIXIT	Protocol Implementation extra Information for Testing

SABTE	Sleep Apnoea Breathing Therapy Equipment	
SDP	Service Discovery Protocol	
SOAP	Simple Object Access Protocol	
TCRL	Test Case Reference List	
TCWG	Test and Certification Working Group	
TP	Test Purpose	
TSS	Test Suite Structure	
URI	Uniform Resource Identifier	
USB	Universal Serial Bus	
WAN	Wide Area Network	
WD	WAN Device	
WDM	Windows Driver Model	
WS	Web Service	
WSDL	Web Service Description Language	
XML	extensible Markup Language	

5 Conventions

The key words "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "MAY", "MAY NOT" in this Recommendation are to be interpreted as in [b-ETSI SR 001 262].

−   SHALL is equivalent to 'must' or 'it is required to'.

−   SHALL NOT is equivalent to 'must not' or 'it is not allowed'.

−   SHOULD is equivalent to 'it is recommended to'.

−   SHOULD NOT is equivalent to 'it is not recommended to'.

−   MAY is equivalent to 'is permitted'.

−   MAY NOT is equivalent to 'it is not required that'.

NOTE – The above-mentioned key words are capitalized for illustrative purposes only and they do not appear capitalized within this Recommendation.

Reference is made in the ITU-T H.800-series of Recommendations to different versions of the Continua Design Guidelines (CDG) by a specific designation. The list of terms that may be used in this Recommendation is provided in Table 1.

Table 1 – List of designations associated with the various versions of the CDG

CDG release	Transposed as	Version	Description	Designation
2016 plus errata	[ITU-T H.810 (2016)]	6.1	Release 2016 plus errata noting all ratified bugs [ITU-T H.810 (2016)].	–
2016	–	6.0	Release 2016 of the CDG including maintenance updates of the CDG 2015 and additional guidelines that cover new functionalities.	Iris
2015 plus errata	[ITU-T H.810 (2015)]	5.1	Release 2015 plus errata noting all ratified bugs [ITU-T H.810 (2015)].	–

Table 1 – List of designations associated with the various versions of the CDG

CDG release	Transposed as	Version	Description	Designation
2015	–	5.0	Release 2015 of the CDG including maintenance updates of the CDG 2013 and additional guidelines that cover new functionalities.	Genome
2013 plus errata	[ITU-T H.810 (2013)]	4.1	Release 2013 plus errata noting all ratified bugs [b-ITU-T H.810 (2013)].	–
2013	–	4.0	Release 2013 of the CDG including maintenance updates of the CDG 2012 and additional guidelines that cover new functionalities.	Endorphin
2012 plus errata	–	3.1	Release 2012 plus errata noting all ratified bugs [b-CDG 2012].	–
2012	–	3.0	Release 2012 of the CDG including maintenance updates of the CDG 2011 and additional guidelines that cover new functionalities.	Catalyst
2011 plus errata	–	2.1	CDG 2011 integrated with identified errata.	–
2011	–	2.0	Release 2011 of the CDG including maintenance updates of the CDG 2010 and additional guidelines that cover new functionalities [b-CDG 2011].	Adrenaline
2010 plus errata	–	1.6	CDG 2010 integrated with identified errata	–
2010	–	1.5	Release 2010 of the CDG with maintenance updates of the CDG Version 1 and additional guidelines that cover new functionalities [b-CDG 2010].	1.5
1.0	–	1.0	First released version of the CDG [b-CDG 1.0].	–

6   Test suite structure (TSS)

The test purposes (TPs) for the WAN interface have been divided into the main subgroups specified below. Annex A describes the TPs for subgroups 2.1.1 to 2.1.3 (shown in bold).

–   Group 1: Sender (SEN)

   •   Group 1.1: Web services interoperability (WSI)

      ○   Subgroup 1.1.1: Basic profile (BP)

      ○   Subgroup 1.1.2: Basic security profile (BSP)

      ○   Subgroup 1.1.3: Reliable messaging (RM)

   •   Group 1.2: SOAP (SOAP)

      ○   Subgroup 1.2.1: SOAP headers (HEAD)

   •   Group 1.3: Audit (ATNA)

      ○   Subgroup 1.3.1: General (GEN)

      ○   Subgroup 1.3.2: PCD-01 (PCD-01)

      ○   Subgroup 1.3.3: Consent management (CM)

- Group 1.4: PCD-01 HL7 messages (PCD-01-DATA)
  - Subgroup 1.4.1: General (GEN)
  - Subgroup 1.4.2: Design guidelines (DG)
  - Subgroup 1.4.3: Pulse oximeter (PO)
  - Subgroup 1.4.4: Blood pressure monitor (BPM)
  - Subgroup 1.4.5: Thermometer (TH)
  - Subgroup 1.4.6: Weighing scales (WEG)
  - Subgroup 1.4.7: Glucose meter (GL)
  - Subgroup 1.4.8: Cardiovascular fitness and activity monitor (CV)
  - Subgroup 1.4.9: Strength fitness equipment (ST)
  - Subgroup 1.4.10: Independent living activity hub (HUB)
  - Subgroup 1.4.11: Adherence monitor (AM)
  - Subgroup 1.4.12: Peak expiratory flow monitor (PF)
  - Subgroup 1.4.13: Body composition analyzer (BCA)
  - Subgroup 1.4.14: Basic electrocardiograph (ECG)
  - Subgroup 1.4.15: International normalized ratio (INR)
  - Subgroup 1.4.16: Sleep apnoea breathing therapy equipment (SABTE)
- Group 1.5: Consent management (CM)
  - Subgroup 1.5.1: WAN XDR transaction (TRANS)
  - Subgroup 1.5.2: WAN metadata validation (META)
  - Subgroup 1.5.3: WAN consent directive validation (CDV)
- Group 1.6: hData observation upload (HDATA)
  - Subgroup 1.6.1: General (GEN)
- Group 1.7: Questionnaires (QUE)
  - Subgroup 1.7.1: General (GEN)
  - Subgroup 1.7.2: CDA validation (CDA)
– Group 2: Receiver (REC)
  - **Group 2.1: Web service interoperability (WSI)**
    - **Subgroup 2.1.1: Basic profile (BP)**
    - **Subgroup 2.1.2: Basic security profile (BSP)**
    - **Subgroup 2.1.3: Reliable messaging (RM)**
  - Group 2.2: SOAP (SOAP)
    - Subgroup 2.2.1: SOAP headers (HEAD)
  - Group 2.3: Audit (ATNA)
    - Subgroup 2.3.1: General (GEN)
    - Subgroup 2.3.2: PCD-01 (PCD-01)
    - Subgroup 2.3.3: Consent management (CM)
  - Group 2.4: PCD-01 HL7 messages (PCD-01-DATA)
    - Subgroup 2.4.1: General (GEN)
    - Subgroup 2.4.2: Design guidelines (DG)

- ○ Subgroup 2.4.3: Pulse oximeter (PO)
- ○ Subgroup 2.4.4: Blood pressure monitor (BPM)
- ○ Subgroup 2.4.5: Thermometer (TH)
- ○ Subgroup 2.4.6: Weighing scales (WEG)
- ○ Subgroup 2.4.7: Glucose meter (GL)
- ○ Subgroup 2.4.8: Cardiovascular fitness and activity monitor (CV)
- ○ Subgroup 2.4.9: Strength fitness equipment (ST)
- ○ Subgroup 2.4.10: Independent living activity hub (HUB)
- ○ Subgroup 2.4.11: Adherence monitor (AM)
- ○ Subgroup 2.4.12: Peak expiratory flow monitor (PF)
- ○ Subgroup 2.4.13: Body composition analyser (BCA)
- ○ Subgroup 2.4.14: Basic electrocardiograph (ECG)
- ○ Subgroup 2.4.15: International normalized ratio (INR)
- ○ Subgroup 2.4.16: Sleep apnoea breathing therapy equipment (SABTE)
- • Group 2.5: Consent management (CM)
  - ○ Subgroup 2.5.1: WAN XDR transaction (TRANS)
  - ○ Subgroup 2.5.2: WAN service validation (SER)
- • Group 2.6: hData observation upload (HDATA)
  - ○ Subgroup 2.6.1: General (GEN)
  - ○ Subgroup 2.6.2: hData record format (HRF)
- • Group 2.7: Questionnaires (QUE)
  - ○ Subgroup 2.7.1: General (GEN)
  - ○ Subgroup 2.7.2: CDA validation (CDA)
  - ○ Subgroup 2.7.3: hData record format (HRF)

7 Electronic attachment

The protocol implementation conformance statements (PICS) and the protocol implementation extra information for testing (PIXIT) required for the implementation of Annex A can be downloaded from http://handle.itu.int/11.1002/2000/12067.

In the electronic attachment, letters "C" and "I" in the column labelled "Mandatory" are used to distinguish between "PICS" and "PIXIT" respectively during testing. If the cell is empty, the corresponding PICS is "independent". If the field contains a "C", the corresponding PICS is dependent on other PICS, and the logical expression is detailed in the "SCR_Expression" field. The static conformance review (SCR) is used in the test tool to assert whether the PICS selection is consistent.

Annex A

Test purposes

(This annex forms an integral part of this Recommendation.)

A.1 TP definition conventions

The test purposes (TPs) are defined according to the following rules:

- **TP Id**: This is a unique identifier (TP/<TT>/<DUT>/<GR>/<SGR>/<XX> – <NNN>). It is specified according to the naming convention defined below:
  - Each test purpose identifier is introduced by the prefix "TP"
  - <TT>: This is the test tool that will be used in the test case.
    - WAN: Wide area network
  - <DUT>: This is the device under test.
    - SEN: WAN observation sender
    - REC: WAN observation receiver
  - <GR>: This identifies a group of test cases.
  - <SGR>: This identifies a subgroup of test cases.
  - <XX>: This identifies the type of testing.
    - BV: Valid behaviour test
    - BI: Invalid behaviour test
  - <NNN>: This is a sequential number that identifies the test purpose (TP).
- **TP label**: This is the title of the TP.
- **Coverage**: This contains the specification reference and clause to be checked by the TP.
  - Spec: This indicates the earliest version of the specification from which the testable items to be checked by the TP are included.
  - Testable item: This contains testable items to be checked by the TP.
- **Test purpose**: This is a description of the requirements to be tested.
- **Applicability**: This contains the PICS items that define if the test case is applicable or not for a specific device. When a TP contains an "ALL" in this field it means that it applies to the device under test within that scope of the test (specialization, transport used, etc.).
- **Other PICS:** This contains additional PICS items (apart from the PICS specified in the Applicability row) which are used within the test case implementation and can modify the final verdict. When this row is empty, it means that only the PICS specified in the Applicability row are used within the test case implementation.
- **Initial condition**: This indicates the state to which the DUT needs to be moved at the beginning of TC execution.
- **Test procedure**: This describes the steps to be followed in order to execute the test case.
- **Pass/Fail criteria**: This provides criteria to decide whether the DUT passes or fails the test case.

A.2   Subgroup 2.1.1 – Basic profile (BP)

TP Id		TP/WAN/REC/WSI/BP/BI-000		
TP label		SOAP Envelope Namespace		
Coverage	Spec	[OASIS/WS-I BP]		
	Testable items	BP-R1015; M		
Test purpose		Check that: A Receiver must generate a fault if they encounter an envelope whose document element is not soap:Envelope		
Applicability		C_REC_000		
Other PICS				
Initial condition		The receiver under test has a WebService enabled and the simulated sender has a SOAP message whose document element is not a soap:Envelope, ready to be sent		
Test procedure		1. The simulated sender sends the SOAP message [b-SOAP 1.2]. 2. The receiver generates a fault.		
Pass/Fail criteria		Check that the receiver generates a fault and does not discard the message.		
Notes				

TP Id		TP/WAN/REC/WSI/BP/BV-000		
TP label		SOAP Envelopes Structure		
Coverage	Spec	[OASIS/WS-I BP]		
	Testable items	BP-R9980; M	BP-R9981; M	BP-R1014; M
		BP-R1008; M	BP-R1009; M	BP-R1033; R
		BP-R1017; M	BP-R1032; M	
Test purpose		Check that: An Envelope must conform to the structure specified in SOAP1.2 Section 5.1, "SOAP Envelope" [AND] an Envelope must have exactly zero or one child elements of the soap:Body element [AND] the children of the soap:body element in an Envelope must be namespace qualified [AND] an Envelope must not contain a Document Type Declaration (DTD) nor Processing Instructions [AND] an Envelope should not contain the namespace declaration xmlns:xml="http//www.w3.org/XML/1998/namespace [AND] a Receiver must not mandate the use of the xsi:type attribute in envelopes except as required in order to indicate a derived type [AND] the soap:envelope, soap:header and soap:body elements in an Envelope must not have attributes in the namespace "http://schemas.xmlsoap.org/soap/envelope/"		
Applicability		C_REC_000		
Other PICS				

Initial condition	The receiver under test has a WebService enabled and the simulated sender has a SOAP message ready to be sent.

Test procedure	1. The simulated sender sends a SOAP message to the receiver under test.
	
	2. The receiver responds with another SOAP message. Check that the captured message has the following structure
	
	    <soap:Envelope *'namespace'*>
	
	        <soap:Header>
	
	            ...
	
	        </soap:Header>
	
	        <soap:Body>
	
	            The children of the soap:envelope are here
	
	        </soap:Body>
	
	    </soap:Envelope>
	
	where soap:Header is optional and it is recommended that the namespace is not "http://www.w3.org/XML/1998/namespace".

Pass/Fail criteria	Check that: • the message contains, in this order, an envelope, an optional header and a body. • each namespace that appears in the soap message is qualified. • the soap:envelope, soap:header and soap:body do not have attributes in the namespace http://schemas.xmlsoap.org/soap/envelope/. • there is no DTD or processing instructions in the envelope. • axsi:type is used only if a derived type is indicated (see XML Schema Part 1: Structures, Section 2.6.1) • the namespace is "http://www.w3.org/2003/05/soap-envelope" to support SOAP 1.2.
Notes	BP-R2201 and BP-R2210 imply that there may be at most one child element of the soap:Body. The referenced errata, NE05, would not be allowed by Continua (as it is not compliant with the WS-I Basic Profile).

TP Id		TP/WAN/REC/WSI/BP/BV-001		
TP label		SOAP encodingStyle Attribute		
Coverage	Spec	[OASIS/WS-I BP]		
	Testable Items	BP-R1005; M	BP-R1006; M	BP-R1007; M
Test purpose		Check that: An Envelope must not contain soap:encodingStyle attributes on any of the elements whose namespace name is "http://schemas.xmlsoap.org/soap/envelope/" [AND] an Envelope must not contain soap:encodingStyle attributes on any element that is a child of soap:body. [AND] an Envelope described in an rpc-literal binding must not contain soap:encodingStyle attribute on any element that is a grandchild of soap:Body		
Applicability		C_REC_000		
Other PICS		C_REC_WSI_003		
Initial condition		The receiver under test has a WebService enabled and the simulated sender has a SOAP message, with a correct soap:encodingStyle attribute in one of the elements, ready to be sent.		

Test procedure	1. The simulated sender sends the SOAP message.
	2. The receiver responds with another SOAP message:
	a. If a soap:encodingStyle attribute is present in any element:
	❑ Namespace is not "http://schemas.xmlsoap.org/soap/envelope/"
	❑ The element is not a child of soap:Body
	❑ If PICS C_REC_WSI_003 is declared, the element is not a grandchild of soap:body
Pass/Fail criteria	In step 2, if the soap:encodingStyle attribute is present, it is as specified.
Notes	

TP Id		TP/WAN/REC/WSI/BP/BV-002		
TP label		Use of SOAP in HTTP		
Coverage	**Spec**	[OASIS/WS-I BP]		
	Testable items	BP-R1127; M	BP-R1140; M	CommonReq1; M
Test purpose		Check that:		
		A Receiver must not rely on the value of the SOAPAction HTTP header to correctly process the message.		
		[AND]		
		A Message shall be sent using HTTP/1.1		
Applicability		C_REC_000		
Other PICS				
Initial condition		The receiver under test has a WebService enabled and the simulated sender has a SOAP message ready to be sent that contains a SOAPAction field not quoted in its HTTP header.		
Test procedure		1. The simulated sender sends a message using HTTP/1.1 with a SOAPAction HTTP Header field not quoted without using security.		
		2. The receiver processes the message (it responds with the fault wsse:InvalidSecurity).		
Pass/Fail criteria		Check that in step 2 the message has been processed.		
Notes				

TP Id		TP/WAN/REC/WSI/BP/BV-003		
TP label		HTTP Status Codes		
Coverage	**Spec**	[OASIS/WS-I BP]		
	Testable items	BP-R1124; M	BP-R1111; R	BP-R1112; R
		BP-R1125; M	BP-R1113; R	BP-R1114; R
		BP-R1115; R		
Test purpose		Check that:		
		An instance must use a 2xx HTTP status code on a response message that indicates the successful outcome of a HTTP request		
		[AND]		
		An instance should use a "200 OK" HTTP status code on a response message that contains an envelope that is not a fault		
		[AND]		
		An Instance should use either a "200 OK" or "202 Accepted" HTTP status code for a response message that does not contain a SOAP envelope but indicates the successful outcome of a HTTP request		
		[AND]		

		An instance must use a 4xx HTTP status code for a response that indicates a problem with the format of a request
		[AND]
		An instance should use a "400 Bad Request" HTTP status code, if a HTTP request message is malformed
		[AND]
		An instance should use a "405 Method not Allowed" HTTP status code if a HTTP request message's method is not "POST".
		[AND]
		An instance should use a "415 Unsupported Media Type" HTTP status code if a HTTP request message's Content-Type header field-value is not permitted by its WSDL description
Applicability		C_REC_000
Other PICS		
Initial condition		The receiver under test has a WebService enabled and the simulated sender is ready to send a HTTP request with an envelope permitted by the SUT
Test procedure		1. The simulated sender sends a HTTP request to the receiver under test with an envelope permitted by the SUT.
		2. The receiver responds with "2xx" as status code. It is recommended to be "200 OK" if the response contains an envelope that is not a fault. It is recommended to be "200 OK" or "202 Accepted", if the response does not contain a SOAP envelope but indicates the successful outcome of the HTTP Request.
		3. The simulated sender sends a HTTP request with a malformed message.
		4. The receiver responds with "4xx" as status code. It is recommended to be "400 Bad Request".
		5. The simulated sender sends a HTTP request with a method that is not "POST"
		6. The receiver responds with "4xx" as status code. It is recommended to be "405 Method not Allowed".
		7. The simulated sender sends a HTTP request with a Content-Type header field not permitted by the receiver's WSDL description.
		8. The Receiver responds with "4xx" as status code. It is recommended to be "415 Unsupported Media Type".
Pass/Fail criteria		Check that status codes are as specified.
Notes		

TP Id		TP/WAN/REC/WSI/BP/BV-004		
TP label		Messages using WSDL descriptions		
Coverage	**Spec**	[OASIS/WS-I BP]		
	Testable items	BP-R2211; M	BP-R2212; M	BP-R2213; M
		BP-R2214; M		
Test purpose		Check that:		
		An Envelope described with an rpc-literal binding must not have the xsi:nil attribute with a value of "1" or "true" on the part accessors		
		[AND]		
		an Envelope must contain exactly one part accessor element for each of the wsdl:parts in the same wsdl:message that are referred to by its soapbind:body element(s)		
		[AND]		
		in a doc-literal description where the value of the parts attribute of soapbind:body is an empty string, the corresponding envelope must have no element content in the soap:Body element		
		[AND]		

		in a rpc-literal description where the value of the parts attribute of soapbind:body is an empty string, the corresponding envelope must have no part accessor elements.
Applicability		C_REC_000 AND (C_REC_WSI_003 OR C_REC_WSI_004)
Other PICS		
Initial condition		The receiver under test has a WebService enabled and the simulated sender is ready to send any SOAP message.

Test procedure		1. The simulated sender sends a SOAP message.
		
		2. The receiver under test responds with another SOAP message.
		
		3. Look into the WSDL of the service and check in the captured message that:
		
		If C_REC_WSI_003:
		
		    a. If the value of the parts attribute of the soapbind:body element of the description is an empty string, there is no part accessor element.
		
		    b. If the value of the parts attribute of the soapbind:body element of the description is not empty, check that the part accessor of the envelope is present and that the value of the xsi:nil attribute, if it is present, is not "1" or "true".
		
		If C_REC_WSI_004:
		
		    a. If the value of the parts attribute of the soapbind:body is an empty string, the envelope does not have element content in the soap:Body element.

Pass/Fail criteria		Check that the envelope is as specified in step 3.
Notes		

TP Id		TP/WAN/REC/WSI/BP/BV-005		
TP label		Port Types		
Coverage	**Spec**	[OASIS/WS-I BP]		
	Testable items	BP-R2301; M		
Test purpose		Check that: The order of the elements in the soap:Body of an envelope must be the same as that of the wsdl:parts in the wsdl:message that describes it for each of the wsdl:part elements bound to the envelope's corresponding soapbind:body element		
Applicability		C_REC_000		
Other PICS				
Initial condition		The receiver under test has a WebService enabled and the simulated sender is ready to send any SOAP message.		
Test procedure		1. The simulated sender sends a SOAP message to the receiver under test. 2. The receiver under test responds with a SOAP message. 3. Check the wsdl:parts elements in the wsdl:message of the WSDL of the receiver under test. 4. Compare them with the soap:Body elements.		
Pass/Fail criteria		In step 4, check that the order of the wsdl:parts are the same as the order of the elements in the soap:Body		
Notes				

TP Id		TP/WAN/REC/WSI/BP/BV-006		
TP label		SOAP Binding		
Coverage	**Spec**	[OASIS/WS-I BP]		
	Testable items	BP-R2742; O	BP-R2743; O	

Test purpose	Check that:
	An envelope may contain fault with a detail element that is not described by a soapbind:fault element in the corresponding WSDL description
	[AND]
	An envelope may contain the details of a header processing related fault in a SOAP header block that is not described by a soapbind:headerfault element in the corresponding WSDL description
Applicability	C_REC_000
Other PICS	
Initial condition	The receiver under test has a WebService enabled and the simulated sender is ready to send a SOAP message that causes a fault in the receiver response.
Test procedure	1. The simulated sender sends a SOAP message that causes a fault at the receiver.
	2. The receiver under test responds with a fault message.
	3. Check the detail element and the SOAP header block.
Pass/Fail criteria	Look into the WSDL description of the web service and check that:
	❑ in step 2, it is optional that the detail element is not described by the soapbind:fault element of the description and that the header block is not described by a soapbind:headerfault element.
Notes	A simulated sender can cause a fault at the receiver in many different ways:
	❑ If the receiver uses security, the sender sends a SOAP envelope without the security header.
	❑ If the sender sends something that is not a SOAP envelope.
	❑ If the receiver uses WSRM, the sender sends something incorrect about the WSRM, such as an unknown sequence, or something like that.

TP Id		TP/WAN/REC/WSI/BP/BV-006_B		
TP label		SOAP Binding 2		
Coverage	Spec	[OASIS/WS-I BP]		
	Testable items	BP-R2712; M	BP-R2729; M	BP-R2735; M
		BP-R2755; M	BP-R2737; M	BP-R2738; M
		BP-R2739; O	BP-R2752; O	BP-R2753; O
Test purpose		Check that:		
		A document-literal binding must be serialized as an envelope with a soap:Body whose child element is an instance of the global element declaration referenced by the corresponding wsdl:message part		
		[AND]		
		An envelope described with an rpc-literal binding that is a response must have a wrapper element whose name is the corresponding wsdl:operation name suffixed with the string "Response".		
		[AND]		
		An envelope described with an rpc-literal binding must place the part accessor elements for parameters and return value in no namespace		
		[AND]		
		The part accessor elements in a message described with an rpc-literal binding must have a local name of the same value as the name attribute of the corresponding wsdl:part element		
		[AND]		
		An envelope described with an rpc-literal binding must namespace qualify the descendents of part accessor elements for the parameters and the return value, as defined by the schema in which the part accessor types are defined		

	[AND]
	An envelope must include all soapbind:headers specified on a wsdl:input or wsdl:output of a wsdl:operation of a wsdl:binding that describes it
	[AND]
	An envelope containing SOAP header blocks that are not described in the appropriate wsdl:binding may have the mustUnderstand attribute on such SOAP header blocks set to '1'.
	[AND]
	An Envelope may contain SOAP header blocks that are not described in the wsdl:binding that describes it
	[AND]
	An envelope may contain more than one instance of each SOAP header block for each soapbind:header element in the appropriate child of soapbind:binding in the corresponding description
Applicability	C_REC_000 AND (C_REC_WSI_003 OR C_REC_WSI_004)
Other PICS	
Initial condition	The receiver under test has a WebService enabled and the simulated sender is ready to send any SOAP message.
Test procedure	1. The simulated sender sends a SOAP message to the receiver under test.
	2. The receiver responds with a SOAP message.
	3. Check the captured message.
Pass/Fail criteria	Look into the WSDL description of the web service and check:
	• In step 2,
	○ If the SOAP header block is not described in wsdl:binding, it can be present and it is optional that the mustUnderstand attribute is present and equal to "1" and that the envelope has more than one instance for each header block.
	○ All soapbind:headers specified in wsdl:input or wsdl:output of a wsdl:operation of a wsdl:binding are included in the envelope.
	○ If C_REC_WSI_003, the part accessor of the envelope has a local name equal to the name of the attribute of the wsdl:part element, it is placed in no namespace and its descendents have a namespace qualified by the schema in which the part accessor types are defined. In addition the envelope has a wrapper element whose name is the corresponding wsdl:operation name suffixed with the string "Response".
	○ If C_REC_WSI_004, the child element of the soap:Body is an instance of the global element declaration referenced by the corresponding wsdl:message part.
Notes	

TP Id		TP/WAN/REC/WSI/BP/BV-007		
TP label		Use of HTTPS		
Coverage	**Spec**	[OASIS/WS-I BP]		
	Testable items	BP-R5000; O	BP-R5001; M	BP-R5010; O
Test purpose		Check that:		
		An Instance may require the use of HTTPS		
		[AND]		
		if an instance requires the use of HTTPS, the location attribute of the soapbind:address element in its wsdl:port description must be a URI whose scheme is "https"; otherwise it must be a URI whose scheme is "http"		
		[AND]		
		an instance may require the use of HTTPS with mutual authentication		

Applicability	C_REC_000
Other PICS	
Initial condition	The receiver under test has a WebService enabled and the simulated sender is ready to send any HTTP request.
Test procedure	1. The simulated sender sends a HTTP request.
	2. Wait until the receiver under test responds using a HTTP instance.
	3. Check the value of the location attribute of the soapbind:address element in its wsdl:port description.
Pass/Fail criteria	In step 2, if this value is "https", the instance requires HTTPS, otherwise, if it is "http", the instance requires HTTP.
Notes	Applicability is ALL because although TI says that HTTPS is optional, the CDG states that HTTPS with TLS must be used.

TP Id		TP/WAN/REC/WSI/BP/BV-008		
TP label		SOAP Processing Model		
Coverage	Spec	[OASIS/WS-I BP]		
	Testable items	BP-R1025; M	BP-R1028; R	BP-R1029; M
		BP-R1030; R	BP-R1027; M	
Test purpose		Check that:		
		A Receiver must handle envelopes in such a way that it appears that all checking of mandatory header blocks is performed before any actual processing		
		[AND]		
		When a fault is generated by a Receiver, further processing should not be performed on the SOAP envelope aside from that which is necessary to rollback, or compensate for, any effects of processing the envelope prior to the generation of the fault.		
		[AND]		
		Where the normal outcome of processing a SOAP envelope would have resulted in the transmission of a SOAP response, but rather a fault is generated instead, a Receiver must transmit a fault in place of the response.		
		[AND]		
		A Receiver that generates a fault should notify the end user that a fault has been generated when practical, by whatever means is deemed appropriate to the circumstance.		
		[AND]		
		A Receiver must generate a "soap:MustUnderstand" fault when an envelope contains a mandatory header block (i.e., one that has a soap:mustUnderstand attribute with the value "1") targeted at the receiver (via soap:actor) that the receiver does not understand.		
Applicability		C_REC_000		
Other PICS				
Initial condition		The receiver under test has a WebService enabled and the simulated sender is ready to send a SOAP message with a header block with soap:MustUnderstand='1'|'true' that the receiver does not understand.		
Test procedure		1. The simulated sender sends the SOAP message.		
		2. The receiver under test generates a soap:MustUnderstand fault.		
		3. The receiver responds with that fault.		
		4. Check that when receiver generates the fault. The simulated sender is notified of the fault by the receiver.		
Pass/Fail criteria		In step 3, the receiver responds with a soap:MustUnderstand fault and no other messages.		
Notes		If the receiver does not send another message besides a soap:MustUnderstand fault, then it is considered that further processing is not performed prior to the generation of the fault.		

TP Id		TP/WAN/REC/WSI/BP/BV-009		
TP label		SOAP Faults		
Coverage	**Spec**	[OASIS/WS-I BP]		
	Testable items	BP-R1107; M	BP-R1002; M	
Test purpose		Check that: A Receiver must interpret a SOAP message as a Fault when the soap:Body of the message has a single soap:Fault child [AND] A Receiver must accept faults that have any number of elements, including zero, appearing as children of the detail element. Such children can be qualified or unqualified		
Applicability		C_REC_000		
Other PICS				
Initial condition		The receiver under test has a WebService enabled and the simulated sender is ready to send a SOAP message with a soap:Fault in the soap:Body.		
Test procedure		1. The simulated sender sends an envelope with a single soap:fault child in the soap:Body. 2. Wait for any response from the receiver under test. 3. The simulated sender sends a soap:fault with zero elements as children of the detail element. 4. Wait for any response of receiver. 5. The simulated sender sends a soap:fault with zero attributes in the detail element. 6. Wait for any response from the receiver.		
Pass/Fail criteria		In step 2,4 and 6 the receiver must not report any error, because all the messages are accepted.		
Notes				

TP Id		TP/WAN/REC/WSI/BP/BV-010		
TP label		WSDL Description		
Coverage	**Spec**	[OASIS/WS-I BP]		
	Testable items	BP-R1034; R	BP-R2028; M	BP-R2029; M
		BP-R4004; M	BP-R4005; R	BP-R4002; R
		BP-R4003; M	BP-R2030; O	BP-R2026; R
		BP-R2101; M	BP-R2102; M	BP-R2105; M
		BP-R2110; M	BP-R2111; M	BP-R2112; R
		BP-R2114; R	BP-R2302; O	BP-R2303; M
		BP-R2304; M	BP-R2305; M	BP-R2709; O
		BP-R2711; R		
Test purpose		Check that: A Description should not contain the namespace declaration xmlns:xml="http://www.w3.org/XML/1998/namespace" [AND] A Description using the WSDL namespace must be valid according to the XML Schema found at "http://ws-i.org/profiles/basic/1.1/wsdl11.xsd". [AND] A Description using the WSDL SOAP binding namespace must be valid according to the XML Schema found at "http://ws-i.org/profiles/basic/1.1/wsdlsoap2004-08-24.xsd"		

[AND]

A Description must use version 1.0 of the eXtensible Markup Language W3C Recommendation

[AND]

A Description may include the Unicode Byte Order Mark (BOM)

[AND]

A Description must use either UTF-8 or UTF-16 encoding

[AND]

In a Description the wsdl:documentation element may be present as the first child element of wsdl:import, wsdl:part and wsdl:definitions in addition to the elements cited in the WSDL1.1 specification

[AND]

A Description should not include extension elements with a wsdl:required attribute value of "true" on any WSDL construct (wsdl:binding, wsdl:portType, wsdl:message, wsdl:types or wsdl:import) that claims conformance to the Profile

[AND]

A Description must not use QName references to WSDL components in namespaces that have been neither imported, nor defined in the referring WSDL document

[AND]

A QName reference to a Schema component in a Description must use the namespace defined in the targetNamespace attribute on the xsd:schema element, or to a namespace defined in the namespace attribute on an xsd:import element within the xsd:schema element

[AND]

All xsd:schema elements contained in a wsdl:types element of a Description must have a targetNamespace attribute with a valid and non-null value, unless the xsd:schema element has xsd:import and/or xsd:annotation as its only child element(s).

[AND]

In a Description, declarations must not extend or restrict the soapenc:Array type

[AND]

In a Description, declarations must not use wsdl:arrayType attribute in the type declaration

[AND]

In a Description, elements should not be named using the convention ArrayOfXXX

[AND]

The target namespace for WSDL definitions and the target namespace for schema definitions in a Description may be the same

[AND]

A Description may use the parameterOrder attribute of an wsdl:operation element to indicate the return value and method signatures as a hint to code generators

[AND]

A Description must not use Solicit-Response and Notification type operations in a wsdl:portType definition

[AND]

A wsdl:portType in a Description must have operations with distinct values for their name attributes

[AND]

A wsdl:operation element child of a wsdl:portType element in a Description must be constructed so that the parameterOrder attribute, if present, omits at most 1 wsdl:part from the output message

[AND]

A wsdl:portType in a Description may have zero or more wsdl:bindings that refer to it, defined

	in the same or other WSDL documents [AND] A Description should not have more than one wsdl:port with the same value for the location attribute of the soapbind:address element
Applicability	C_REC_000
Other PICS	
Initial condition	The receiver has published its WSDL description.

Test procedure	1. Look up the WSDL description using the corresponding URL given by the receiver under test. Check that:
	
	   a. xmlns:xml≠ "http://www.w3.org/XML/1998/namespace"
	
	   b. XML version = "1.0"
	
	   c. UTF-8 or UTF-16 encoding are used and the unicode byte order mark (BOM) is optional.
	
	   d. if wsdl:documentation is present, check that it is the first child element of wsdl:import, wsdl:part or wsdl:definitions.
	
	   e. the targetNamespace attribute of an xsd:schema contained in wsdl:types element, has a valid non-null value, unless the xsd:schema has xsd:import and/or xsd:annotation as its only child element(s)
	
	    <types>
	     <xsd:schema **targetNamespace**="http://example.org/foo/"
	     … >
	
	   f. the wsdl:portType definition does not use Solicit-Response or Notification Type operations and has operations with distinct values for their name attributes:
	
	    <**portType** name="BarPortType">
	     <operation name="BarOperation">
	     <input message="bar:BarMsg"/>
	     </operation>
	    </portType>
	
	   g. if present the parameterOrder attribute of the wsdl:operation, that is a child of wsdl:portType, omits at most 1 wsdl:part from the output message.
	
	   h. the wsdl:ArrayType is not present on type declaration.
	
	   i. the soapenc:ArrayType is not extended or restricted.
	
	   j. the description does not contain any extension elements with a wsdl:required attribute value of "true" on any WSDL construct (wsdl:binding, wsdl:portType, wsdl:message, wsdl:types or wsdl:import) as is recommended.
	
	   k. The namespace of a QName reference to a schema component is defined in the targetNamespace attribute on the xsd:schema element or is the namespace of the xsd:import element.
	
	   l. aQName reference to WSDL components in namespaces that have been neither imported nor defined on the referring WSDL Document, is not used.
	
	   m. wsdl:bindings are optional
	
	   n. that more than one wsdl:port with the same value for the location attribute of the soapbind:address element are not used as is recommended.

Pass/Fail criteria	Check that: • in step 1, the sender can access the WSDL description. • all elements and attributes are as specified. • the description using the wsdl namespace is valid according to the XML schema found at http://ws-i.org/profiles/basic/1.1/wsdl11.xsd. • the description using the WSDL SOAP Bind namespace is valid according to the XML schema found at http://ws-i.org/profiles/basic/1.1/wsdlsoap-2004-08-24.xsd.

Notes		BP-R4005 is the same that BP-R1034		

TP Id		TP/WAN/REC/WSI/BP/BV-011		
TP label		WSDL Description: wsdl:binding		
Coverage	Spec	[OASIS/WS-I BP]		
	Testable items	BP-R2209; R	BP-R2202; O	BP-R2208; O
		BP-R2205; M	BP-R2701; M	BP-R2702; M
		BP-R2705; M	BP-R2706; M	BP-R2710; M
		BP-R2716; M	BP-R2717; M	BP-R2726; M
		BP-R2718; M	BP-R2719; O	BP-R2740; R
		BP-R2741; R	BP-R2720; M	BP-R2749; M
		BP-R2721; M	BP-R2754; M	BP-R2722; O
		BP-R2723; M	BP-R2707; M	BP-R2751; M
Test purpose		Check that: A wsdl:binding in a Description should bind every wsdl:part of a wsdl:message in the wsdl:portType to which it refers with a binding extension element. [AND] A wsdl:binding in a Description may contain soapbind:body element(s) that specify that zero parts form the soap:Body [AND] A binding in a Description may contain soapbind:header element(s) that refer to wsdl:parts in the same wsdl:message that are referred to by its soapbind:body element(s). [AND] A wsdl:binding in a Description must refer, in each of its soapbind:header, soapbind:headerfault and soapbind:fault elements, only to wsdl:part element(s) that have been defined using the element attribute [AND] The wsdl:binding element in a Description must be constructed so that its soapbind:binding child element specifies the transport attribute. [AND] A wsdl:binding element in a Description must specify the HTTP transport protocol with SOAP binding. Specifically, the transport attribute of its soapbind:binding child must have the value "http://schemas.xmlsoap.org/soap/http". [AND] A wsdl:binding in a Description must either be a rpc-literal binding or a document-literal binding. [AND] The operations in a wsdl:binding in a Description must result in operation signatures that are different from one another [AND] A wsdl:binding in a Description must use the value of "literal" for the use attribute in all soapbind:body, soapbind:fault, soapbind:header and soapbind:headerfault elements [AND] A document-literal binding in a Description must not have the namespace attribute specified on contained soapbind:body, soapbind:header, soapbind:headerfault and soapbind:fault elements [AND] An rpc-literal binding in a Description must have the namespace attribute specified, the value of which must be an absolute URI, on contained soapbind:body elements

	[AND]
	An rpc-literal binding in a Description must not have the namespace attribute specified on contained soapbind:header, soapbind:headerfault and soapbind:fault elements
	[AND]
	A wsdl:binding in a Description must have the same set of wsdl:operations as the wsdl:portType to which it refers
	[AND]
	A wsdl:binding in a Description may contain no soapbind:headerfault elements if there are no known header faults
	[AND]
	A wsdl:binding in a Description should contain a soapbind:fault describing each known fault
	[AND]
	A wsdl:binding in a Description should contain a soapbind:headerfault describing each known header fault
	[AND]
	A wsdl:binding in a Description must use the part attribute with a schema type of "NMTOKEN" on all contained soapbind:header and soapbind:headerfault elements
	[AND]
	A wsdl:binding in a Description must not use the parts attribute on contained soapbind:header and soapbind:headerfault elements
	[AND]
	A wsdl:binding in a Description must have the name attribute specified on all contained soapbind:fault elements
	[AND]
	In a Description, the value of the name attribute on a soapbind:fault element must match the value of the name attribute on its parent wsdl:fault element
	[AND]
	A wsdl:binding in a Description may specify the use attribute on contained soapbind:fault elements
	[AND]
	If in a wsdl:binding in a Description the use attribute on a contained soapbind:fault element is present, its value must be "literal".
	[AND]
	A wsdl:binding in a Description that contains one or more soapbind:body, soapbind:fault, soapbind:header or soapbind:headerfault elements that do not specify the use attribute must be interpreted as though the value "literal" had been specified in each case
	[AND]
	The order of soapbind:header elements in soapbind:binding sections of a Description must be considered independent of the order of SOAP header blocks in the envelope
Applicability	C_REC_000 AND (C_REC_WSI_003 OR C_REC_WSI_004)
Other PICS	
Initial condition	The receiver has published its WSDL description.

Test procedure	1. Look up the WSDL description using the corresponding URL given by the receiver under test. If wsdl:binding is present, check that:
	
	    a. the Soapbind:binding child element specifies the transport attribute and transport="http://schemas.xmlsoap.org/soap/http".
	
	    b. the soapbind:header and soapbind:body elements are optional.
	
	    c. the wsdl:binding refers in soapbind:headerfault, soapbind:header, soapbind:fault elements only to wsdl:parts that has been defined using the element attribute.
	
	    d. the operations resulted in operation signatures that are different from one

another.

    e.   the "use" attribute in soapbind:header, soapbind:body, soapbind:headerfault and soapbidn:fault, if they are present, is "literal".

    f.   the wsdl:binding has the same wsdl:operations as wsdl:portType.

    g.   the part attribute of soapbind:header and soapbind:headerfault elements, if they are present, have the schema type of "NMTOKEN".

    h.   the soapbind:headerfault elements are optional if there are no known header faults.

    i.   all soapbind:fault elements have the name attribute specified and its value matches the value of the name attribute on its parent wsdl:fault element. The "use" attribute is optional.

    j.   the order of the soapbind:header element, if it is present, is independent of the order of SOAP header blocks.

    k.   if C_REC_WSI_003, the namespace attribute is specified only on a contained soapbind:body and its value is an absolute URI.

        if C_REC_WSI_004, the namespace attribute is not specified.

Pass/Fail criteria	The sender can access the WSDL description and if wsdl:binding is present, elements and attributes are as specified above.
Notes	The profile defines the "operation signature" to be the fully qualified name of the child element of SOAP body of the SOAP input message described by an operation in a WSDL binding. In the case of rpc-literal binding, the operation name is used as a wrapper for the part accessors. In the document-literal case, since a wrapper with the operation name is not present, the message signatures must be correctly designed.

TP Id		TP/WAN/REC/WSI/BP/BV-012		
TP label		WSDL Description. Imported Descriptions		
Coverage	Spec	[OASIS/WS-I BP]		
	Testable items	BP-R2001; M	BP-R2803; M	BP-R2002; M
		BP-R2003; M	BP-R2004; M	BP-R2009; O
		BP-R2010; M	BP-R2011; M	BP-R2007; M
		BP-R2022; M	BP-R2023; M	BP-R2005; M
Test purpose		Check that: A Description must only use the WSDL "import" statement to import another WSDL description [AND] In a Description, the namespace attribute of the wsdl:import must not be a relative URI [AND] To import XML Schema Definitions, a Description must use the XML Schema "import" statement [AND] A Description must use the XML Schema "import" statement only within the xsd:schema element of the types section [AND] In a Description the schemaLocation attribute of an xsd:import element must not resolve to any document whose root element is not "schema" from the namespace "http://www.w3.org/2001/XMLSchema" [AND] An XML Schema directly or indirectly imported by a Description may include the Unicode Byte Order Mark (BOM). [AND]		

		An XML Schema directly or indirectly imported by a Description must use either UTF-8 or UTF-16 encoding
		[AND]
		An XML Schema directly or indirectly imported by a Description must use version 1.0 of the eXtensible Markup Language W3C Recommendation
		[AND]
		A Description must specify a non-empty location attribute on the wsdl:import element
		[AND]
		When they appear in a Description, wsdl:import elements must precede all other elements from the WSDL namespace except wsdl:documentation
		[AND]
		When they appear in a Description, wsdl:types elements must precede all other elements from the WSDL namespace except wsdl:documentation and wsdl:import
		[AND]
		The targetNamespace attribute on the wsdl:definitions element of a description that is being imported must have same the value as the namespace attribute on the wsdl:import element in the importing Description
Applicability		C_REC_000 AND C_REC_WSI_002
Other PICS		
Initial condition		The receiver has published its WSDL description.
Test procedure		1. Look up the WSDL description using the corresponding URL given by the receiver under test. If the wsdl:import element is present, check that:
		a. the wsdl:import is only used to import another wsdl description.
		b. the namespace of the wsdl:import is not a relative URI.
		c. the XML schema "import" statement is used to import the XML schema definitions within the xsd:schema element.
		d. an imported XML schema definitions is version 1.0.
		e. the schemaLocation attribute of the xsd:import element is resolved to a document whose root element is a schema from the namespace "http://www.w3.org/2001/XMLSchema"
		f. UTF-8 or UTF-16 encoding is used and it is optional that it includes the unicode byte order mark (BOM).
		g. the location attribute of the wsdl:import element is not empty.
		h. the wsdl:import precedes all other elements from the WSDL, except wsdl:documentation.
		i. wsdl:types precedes all other elements from the WSDL, except wsdl:documentation and wsdl:import.
		j. the targetNamespace attribute of the description being imported is the same as the namespace attribute on the wsdl:import element in the importing description.
Pass/Fail criteria		The sender can access the WSDL description and that elements and attributes of wsdl:import are as specified above.
Notes		

TP Id		TP/WAN/REC/WSI/BP/BV-013		
TP label		WSDL Description: wsdl:parts element		
Coverage	Spec	[OASIS/WS-I BP]		
	Testable items	BP-R2201; C	BP-R2210; C	BP-R2203; C
		BP-R2207; O	BP-R2204; C	BP-R2206; M
		BP-R2306; M		

Test purpose	Check that:
	A document-literal binding in a Description must, in each of its soapbind:body element(s), have at most one part listed in the parts attribute, if the parts attribute is specified
	[AND]
	If a document-literal binding in a Description does not specify the parts attribute on a soapbind:body element, the corresponding abstract wsdl:message must define zero or one wsdl:parts
	[AND]
	If a rpc-literal binding in a Description must refer, in its soapbind:body element(s), only to wsdl:part element(s) that have been defined using the type attribute
	[AND]
	A wsdl:message in a Description may contain wsdl:parts that use the elements attribute provided those wsdl:parts are not referred to by a soapbind:body in an rpc-literal binding
	[AND]
	A document-literal binding in a Description must refer, in each of its soapbind:body element(s), only to wsdl:part element(s) that have been defined using the element attribute
	[AND]
	A wsdl:message in a Description containing a wsdl:part that uses the element attribute must refer, in that attribute, to a global element declaration
	[AND]
	A wsdl:message in a Description must not specify both type and element attributes on the same wsdl:part
Applicability	C_REC_000 AND (C_REC_WSI_003 OR C_REC_WSI_004)
Other PICS	
Initial condition	The receiver has published its WSDL description.

Test procedure	1. Look up the WSDL description using the corresponding URL given by the receiver under test.
	
	An example of a part element in a description is shown below:
	
	    `<message name="GetTradePriceInput">`
	     `<part name="body" element="tns:SubscribeToQuotes"/>`
	    `</message>`

Pass/Fail criteria	Check that: • if C_REC_WSI_004 is supported and the receiver does not specify the parts attribute on a soapbind:body element, the wsdl:message defines zero or one wsdl:parts. o If the receiver does specify the doc-literal binding, it has at most one part listed in the parts attribute and it is defined using the element attribute, that refers to a global element declaration. • if C_REC_WSI_003 is supported, the receiver refers in its soapbind:body element(s) only to a wsdl:part element(s) defined using the type attribute. wsdl:parts that uses the elements attribute and this provided those wsdl:parts are not referred to by a soapbind:body are optional. • in either case above, the wsdl:message does not specify both type and element attributes on the same wsdl:part.
Notes	

A.3   Subgroup 2.1.2 – Basic security profile (BSP)

TP Id		TP/WAN/REC/WSI/BSP/BV-000		
TP Label		TLS and SSL		
Coverage	Spec	[OASIS/WS-I BSP]		
	Testable items	BSP-322; R	BSP-323; R	
	Spec	[b-CDG 2012], WAN interface		
	Testable items	SecGuidelines2; M		
Test purpose		Check that:		
		As the AES encryption algorithm is intended to supersede the 3DES algorithm, it is recommended that TLS-capable implementations implement TLS_RSA_WITH_AES_128_CBC_SHA or the FIPS equivalent		
		[AND]		
		The ciphersuites defined in the TLS specifications that use anonymous Diffie-Hellman (i.e. those that have DH_anon in their symbolic name) are vulnerable to man-in-the-middle attacks. It is also recommended that ciphersuites that include MD5 (i.e. those that have MD5 in their symbolic name) be avoided, due to known security weaknesses of the MD5 algorithm. It is recommended that such ciphersuites be avoided.		
		The Profile recommends against the use of the following ciphersuites due to their lack of confidentiality services:		
		•TLS_RSA_WITH_NULL_SHA		
		•TLS_RSA_WITH_NULL_MD5		
		It is also recommended that ciphersuites that use 40 or 56 bit keys be avoided, due to their relative ease of compromise through brute-force attack.		
		[AND]		
		Continua WAN client and service components shall support AES cipher as specified in RFC 3268.		
Applicability		C_REC_000		
Other PICS		C_REC_WSI_005, C_REC_WSI_029, C_REC_WSI_030, C_REC_WSI_031, C_REC_WSI_032		
Initial condition		The simulated sender and the receiver under test have never been partners in a message exchange.		
Test procedure		1. If instance is FIPS compliant (C_REC_WSI_005=true):		

1. If instance is FIPS compliant (C_REC_WSI_005=true):

   a. Load the simulated sender supporting TLS_RSA_FIPS_WITH_AES_128_CBC_SHA

   b. Make the receiver under test establish a TLS connection.

   c. Check in TLS handshake that the receiver under test SHOULD not support:

   ❑ Any cipher-suites with an DH_anon in their symbolic name

   ❑ Any cipher-suites with a MD5 in their symbolic name

   ❑ Any of the following cipher-suites:

   • TLS_RSA_WITH_NULL_SHA

   • TLS_RSA_WITH_NULL_MD5

   ❑ Any cipher-suites that use 40 or 56 bit keys

   d. Check that the receiver under test supports TLS_RSA_FIPS_WITH_AES_128_CBC_SHA

   e. Close the connection.

2. If an instance is not FIPS compliant (C_REC_WSI_005=false):

   a. Load the simulated sender supporting TLS_RSA_WITH_AES_128_CBC_SHA.

<table>
<tr><td></td><td colspan="3">b. Make the receiver under test establish a TLS connection.

c. Check in the TLS handshake that the receiver under test SHOULD not support:

    ❑ any cipher-suites with a DH_anon in their symbolic name.

    ❑ any cipher-suites with a MD5 in their symbolic name.

    ❑ any of the following cipher-suites:

        • TLS_RSA_WITH_NULL_SHA

        • TLS_RSA_WITH_NULL_MD5

    ❑ any cipher-suites that use 40 or 56 bit keys.

d. Check that the receiver under test MUST support TLS_RSA_WITH_AES_128_CBC_SHA.

e. Close the connection.</td></tr>
<tr><td>Pass/Fail criteria</td><td colspan="3">• If C_REC_WSI_005, the receiver under test must support TLS_RSA_FIPS_WITH_AES_128_CBC_SHA.

• If not C_REC_WSI_005, the receiver under test must support TLS_RSA_WITH_AES_128_CBC_SHA.

• The cipher-suites supported must match with these PICS: C_REC_WSI_029, C_REC_WSI_030, C_REC_WSI_031, C_REC_WSI_032.</td></tr>
<tr><td>Notes</td><td colspan="3"></td></tr>
</table>

<table>
<tr><td>TP Id</td><td></td><td colspan="3">TP/WAN/REC/WSI/BSP/BV-003</td></tr>
<tr><td>TP label</td><td></td><td colspan="3">Basic Profile Clarification</td></tr>
<tr><td rowspan="4">Coverage</td><td>Spec</td><td colspan="3">[OASIS/WS-I BSP]</td></tr>
<tr><td rowspan="3">Testable items</td><td>BSP-R5814; O</td><td>BSP-R5801; M</td><td>BSP-R5803; M</td></tr>
<tr><td>BSP-R5805; M</td><td>BSP-R5807; M</td><td>BSP-R5809; M</td></tr>
<tr><td>BSP-R5811; M</td><td>BSP-R5813; M</td><td></td></tr>
<tr><td>Test purpose</td><td colspan="4">Check that:

Where the normal outcome of processing a SECURE_ENVELOPE would have resulted in the transmission of a SOAP Response, but rather a fault is generated instead, a Receiver may transmit a fault or silently discard the message

[AND]

bp11:R2301 must be true after any SOAP Message Security has been reversed for the Envelope

[AND]

bp11:R2710 must be true after SOAP Message Security processing has been reversed for the Envelope

[AND]

bp11:R2712 must be true after any SOAP Message Security has been reversed for the Envelope

[AND]

For bp11:R2724 "Inconsistent" must be taken to mean "Inconsistent after SOAP Message security has been reversed", for the Envelope

[AND]

With respect to bp11:R2725 the Instance must check for consistency of the Envelope per BP 1.1 after reversing SOAP Message Security

[AND]

With respect to bp11:R2729 the verification of the wrapper element name of the Envelope must be performed after reversing SOAP Message Security.

[AND]</td></tr>
</table>

		With respect to bp11:R2738 verification of an Envelope must occur after SOAP Message Security has been reversed
Applicability		C_REC_000 AND C_REC_WSI_006
Other PICS		C_REC_WSI_003, C_REC_WSI_004
Initial condition		The receiver under test has a WebService enabled and the simulated sender is ready to send a SOAP message using the same security policy as the receiver.

Test procedure		1. The simulated sender sends a message using SOAP message security.
		
		2. The receiver under test responds using SOAP message security.
		
		3. The simulated sender takes the WSDL description and after reversing the SOAP message security of the response, check that:
		
		    a. the order of the elements in the soap:body is the same as that of the wsdl:parts in the wsdl:message.
		
		    b. operations in wsdl:binding result in operations signatures that are different from one another.
		
		    c. the envelope includes all the soapbind:headers specified on a wsdl:input or wsdl:output of a wsdl:operation of a wsdl:binding.
		
		    d. if C_REC_WSI_003, the envelope has a wrapper element whose name is the corresponding wsdl:operation name suffixed with the string "Response".
		
		    e. if C_REC_WSI_004, the binding is serialized as an envelope with a soap:Body whose child element is an instance of the global element declaration referenced by the corresponding wsdl:message part.
		
		4. The simulated sender sends an envelope with an incorrect namespace.
		
		5. The receiver generates a soap:Fault with a faultcode= "VersionMismatch".
		
		6. The simulated sender sends an envelope with an incorrect namespace and a soap:MustUnderstand attribute value of "1".
		
		7. The receiver generates a soap:Fault with a faultcode= "VersionMismatch".
		
		8. The simulated sender sends an envelope with a correct namespace and soap:MustUnderstand attribute value of "1" using security that the receiver is not going to understand.
		
		9. The receiver generates a soap:Fault with a faultcode="MustUnderstand".
		
		10. The simulated sender sends an envelope with a correct namespace, soap:MustUnderstand attribute value of "0" and that is inconsistent with its WSDL description.
		
		11. The receiver generates a soap:Fault with a faultcode="Sender".

Pass/Fail criteria		All steps are as specified. When the receiver generates a soap:Fault, it can transmit it or discard the message.
Notes		"Reverse" means to remove impacts of applying SOAP message security that has been applied to an envelope created according to BP 1.1 • bp11:R1029 states "Where the normal outcome of processing a SOAP Envelope would have resulted in the transmission of a SOAP response, but rather a fault is generated instead, a RECEIVER MUST transmit a fault place of the response" • bp11:R2301 states "The order of the elements in the soap:body of an ENVELOPE MUST be the same as that of the wsdl:parts in the wsdl:message that describes it." • bp11:R2710 states "The operations in a wsdl:binding in a DESCRIPTION MUST result in operation signatures that are different from one another." • bp11:R2712 states "A document-literal binding MUST be serialized as an ENVELOPE with a soap:Body whose child element is an instance of the global element declaration referenced by the corresponding wsdl:message part." • bp11:R2724 states "If an INSTANCE receives an envelope that is inconsistent with its WSDL description, it SHOULD generate a soap:Fault with a faultcode of 'Client', unless a 'MustUnderstand' or 'VersionMismatch' fault is generated." • bp11:R2725 states "If an INSTANCE receives an envelope that is inconsistent with its WSDL description, it MUST check for "VersionMismatch", "MustUnderstand" and "Client"

		fault conditions in that order."
		• bp11:R2729 states "An ENVELOPE described with an rpc-literal binding that is a response MUST have a wrapper element whose name is the corresponding wsdl:operation name suffixed with the string 'Response'."
		• bp11:R2738 states "An ENVELOPE MUST include all soapbind:headers specified on a wsdl:input or wsdl:output of a wsdl:operation of a wsdl:binding that describes it."

TP Id		TP/WAN/REC/WSI/BSP/BV-004		
TP label		Timestamp element		
Coverage	Spec	[OASIS/WS-I BSP]		
	Testable items	BSP-R3227; M	BSP-R3203; M	BSP-R3224; R
		BSP-R3221; M	BSP-R3222; M	BSP-R3220; R
		BSP-R3229; R	BSP-R3213; M	BSP-R3215; M
		BSP-R3225; M	BSP-R3226; M	BSP-R3217; M
		BSP-R3223; M		
Test purpose		Check that:		
		A SECURITY_HEADER must not contain more than one Timestamp		
		[AND]		
		A Timestamp must contain exactly one Created		
		[AND]		
		Any Timestamp must not contain more than one Expires		
		[AND]		
		Any Timestamp containing an Expires must contain a Created that precedes its sibling Expires		
		[AND]		
		Any Timestamp must not contain anything other than Created or Expires elements		
		[AND]		
		Any Created should not contain a seconds value with more than three digits to the right of the decimal (milliseconds).		
		[AND]		
		Any Expires should not contain a seconds value with more than three digits to the right of the decimal (milliseconds).		
		[AND]		
		Any Created containing second values must specify seconds values less than 60		
		[AND]		
		Any Expires containing second values must specify seconds values less than 60		
		[AND]		
		Any Created must not include a ValueType attribute		
		[AND]		
		Any Expires must not include a ValueType attribute		
		[AND]		
		Any Created must contain time values in UTC format as specified by the XML Schema type (dateTime).		
		[AND]		
		Any Expired must contain time values in UTC format as specified by the XML Schema type (dateTime).		
Applicability		C_REC_000 AND C_REC_WSI_007		

Other PICS	
Initial condition	The receiver under test has a WebService enabled and the simulated sender is ready to send a SOAP message with a Timestamp

Test procedure	1. The simulated sender sends the message using a Timestamp element.
	
	2. The receiver under test responds to the message.
	
	3. Check in the response message that:
	
	    a. the Timestamp is present and there is only one. For example:
	
	        <wsu:Timestamp wsu:Id="*timestamp*">
	
	         <wsu:Created>*2001-09-13T08:42:00Z*</wsu:Created>
	
	         <wsu:Expires>*2001-10-13T09:00:00Z*</wsu:Expires>
	
	        </wsu:Timestamp>
	
	    b. only one created element is present and inside it:
	
	      ❑ ValueType attribute is not included
	
	      ❑ UTC format is used in time values
	
	      ❑ seconds values are less than 60 and its decimal values are recommended to be less than 3 digits to the right
	
	    c. if an Expires element is present, there is only one and it comes after the created element and:
	
	      ❑ ValueType attribute is not included
	
	      ❑ UTC format is used in time values
	
	      ❑ the seconds values are less than 60 and its decimal values are recommended to be less than 3 digits to the right

Pass/Fail criteria	The elements in step 3 are as specified.
Notes	

TP Id		TP/WAN/REC/WSI/BSP/BV-005		
TP label		Security Token References - Direct References		
Coverage	Spec	[OASIS/WS-I BSP]		
	Testable items	BSP-R3061; M	BSP-R3057; M	BSP-R3064; M
		BSP-R3059; M	BSP-R3058; M	BSP-R3062; M
		BSP-R3027; M	BSP-R3211; M	
Test purpose		Check that: A SECURITY_TOKEN_REFERENCE must provide exactly one token reference [AND] Any STR_REFERENCE must not reference a SECURITY_TOKEN_REFERENCE [AND] Any STR_REFERENCE must not reference an STR_EMBEDDED [AND] Any STR_REFERENCE must specify a ValueType attribute [AND] Any STR_REFERENCE ValueType attribute must contain a value for the referenced SECURITY_TOKEN specified by the corresponding security token profile. [AND] Any STR_REFERENCE must specify a URI attribute		

	[AND]
	Any SECURITY_TOKEN_REFERENCE must not contain an STR_KEY_NAME
	[AND]
	Any SECURITY_TOKEN_REFERENCE must not reference a ds:KeyInfo element
Applicability	C_REC_000 AND C_REC_WSI_019
Other PICS	
Initial condition	The receiver under test has a WebService enabled and the simulated sender is ready to send a SOAP message with the same security policy as the receiver.
Test procedure	1. The simulated sender sends a message using a security token reference (STR) with an STR reference.
	2. The receiver under test responds with a message including a SecurityTokenReference with a direct reference:
	`<wsse:SecurityTokenReference wsu:Id="...">`
	`<wsse:Reference URI="..." ValueType="..."/>`
	`</wsse:SecurityTokenReference>`
	3. Check in the captured message that:
	a. there is only one STR_Reference to the SECURITY_TOKEN_REFERENCE
	b. the STR_Reference does not reference another STR or an STR_Embedded.
	c. a URI Attribute is present.
	d. a ValueType attribute is present and it contains a value for the referenced security token specified by the corresponding security token profile (e.g., an X.509 certificate token).
	e. the STR does not contain an STR_KEY_NAME and does not reference a ds:KeyInfo element.
Pass/Fail criteria	Check that the STR is as specified in steps 2 and 3.
Notes	

TP Id		TP/WAN/REC/WSI/BSP/BV-006		
TP label		Security Token References - Key Identifier References		
Coverage	**Spec**	[OASIS/WS-I BSP]		
	Testable items	BSP-R3054; M	BSP-R3063; M	BSP-R3070; M
		BSP-R3071; M		
Test purpose		Check that:		
		Any STR_KEY_IDENTIFIER must specify a ValueType attribute		
		[AND]		
		Any STR_KEY_IDENTIFIER ValueType attribute must contain a value specified within the security token profile associated with the referenced SECURITY_TOKEN		
		[AND]		
		Any STR_KEY_IDENTIFIER that refers to a SECURITY_TOKEN other than a SAML_TOKEN must specify an EncodingType attribute		
		[AND]		
		Any STR_KEY_IDENTIFIER EncodingType attribute must have a value of "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary".		
Applicability		C_REC_000 AND C_REC_WSI_020		
Other PICS				
Initial condition		The receiver under test has a WebService enabled and the simulated sender is ready to send a SOAP message with the same security policy as the receiver.		

Test procedure	1.	The simulated sender sends a message using a security token reference (STR) with a key identifier reference.
	2.	The receiver under test responds with a message including a SecurityTokenReference with a key identifier reference:
		<wsse:SecurityTokenReference>
		<wsse:KeyIdentifier wsu:Id="..."
		ValueType="..."
		EncodingType="...">
		...
		</wsse:KeyIdentifier>
		</wsse:SecurityTokenReference>
	3.	Check in the captured message that:
		a. ValueType is present and contains a value specified within the security token profile associated with the referenced security token.
		b. if SAML Token is referenced, an encodingType attribute is not present. If SAML Token is not referenced, encodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary".
Pass/Fail criteria	In step 3, the attributes are as specified.	
Notes		

TP Id		TP/WAN/REC/WSI/BSP/BV-024		
TP label		SoapAction Header		
Coverage	Spec	[OASIS/WS-I BSP]		
	Testable items	BSP-C2010; R		
Test purpose		Check that: In a Description, the soapAction attibute of a soapbind:operation element should be either omitted, or have as its value an empty string		
Applicability		C_REC_000		
Other PICS				
Initial condition		The receiver under test has a WebService enabled and its WSDL description is available.		
Test procedure		1. Take the wsdl description using the URL provided by the receiver under test (I_REC_WSI_001) 2. Check that in soapbind:operation element, the soapAction attribute is omitted or its value is an empty string..		
Pass/Fail criteria		In step 2, it is recommended that the soapAction attribute is omitted or that its value is an empty string, if it is present and includes any value, a warning is issued.		
Notes		This test case verifies a recommended behaviour and therefore it will never result is a fail.		

A.4   Subgroup 2.1.3 – Reliable messaging (RM)

TP Id		TP/WAN/REC/WSI/RM/BV-000_A		
TP label		Protocol Preconditions		
Coverage	Spec	[OASIS WS-I RM]		
	Testable items	Namespace; M		
Test purpose		Check that: The XML namespace URI that MUST be used by implementations of this specification is: http://docs.oasis-open.org/ws-rx/wsrm/200702		
Applicability		C_REC_000		
Other PICS				
Initial condition		The simulated sender and the receiver under test are in the none sequence state.		
Test procedure		1. The simulated sender sends a CreateSequence with an offer element to the receiver. 2. The receiver under test responds with a CreateSequenceResponse accepting the offer. 3. The simulated sender sends a sequence. 4. The receiver under test responds with its sequence and a SequenceAcknowledgement element. 5. The simulated sender sends a SequenceAcknowledgement element.		
Pass/Fail criteria		Check that in every wsrm element its XML namespace is: xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrm/200702",		
Notes				

TP Id		TP/WAN/REC/WSI/RM/BV-000_B		
TP label		Delivery Assurances		
Coverage	Spec	[OASIS WS-I RM]		
	Testable items	DelivAssurance 4; C	DelivAssurance 7; C	
	Spec	[b-CDG 2012], WAN Interface		
	Testable items	CommonReq 2; O	CommonReq 3; R	
Test purpose		Check that: AtMostOnce assertion sets that each message is to be delivered at most once. The RM Source MAY retry transmission of unacknowledged messages, but is NOT REQUIRED to do so. [AND] The requirement on an RM Source using ExactlyOnce assertion is that it SHOULD retry transmission of every message sent by the Application Source until it receives an acknowledgement from the RM Destination [AND] Continua WAN client and service components may transmit messages from the Continua better QoS bin using a WS-ReliableMessaging sequence configured to use 'AtMostOnce' message delivery. [AND] Continua WAN client and service components should transmit messages from the Continua best QoS bin using a WS-ReliableMessaging sequence configured to use 'ExactlyOnce' message delivery.		

Applicability	C_REC_000 AND (C_REC_WSI_025 OR C_REC_WSI_026)
Other PICS	
Initial condition	The simulated sender and the receiver under test are in the none sequence state.
Test procedure	1. The simulated sender sends a CreateSequence message with an offer element.
	2. The receiver under test responds with CreateSequenceResponse.
	3. The sender sends a sequence message indicating that it is the last message. Note that when the receiver acknowledges that sequence, the sender and receiver switch WSRM roles: the sender becomes an RM destination and the receiver becomes an RM source.
	4. The receiver responds with the SequenceAcknowledgement and a sequence message indicating that it is the last message.
	5. The sender does not send the SequenceAcknowledgement.
	6. If C_REC_WSI_025, the receiver may retry transmission.
	7. If C_REC_WSI_026, the receiver should retry transmission.
Pass/Fail criteria	All steps are as indicated.
Notes	

TP Id		TP/WAN/REC/WSI/RM/BV-001		
TP label		Considerations on the Use of Extensibility Points		
Coverage	Spec	[OASIS WS-I RM]		
	Testable items	ExtensPoints 2; R		
Test purpose		Check that:		
		If a receiver does not recognize an extension, the receiver SHOULD ignore the extension		
Applicability		C_REC_000		
Other PICS				
Initial condition		The simulated sender and the receiver under test are in the none sequence state.		
Test procedure		1. The simulated sender sends a CreateSequence message using an extensibility point that the receiver does not recognize, such as <myExtensibilityPoint />.		
		2. The receiver under test should respond, ignoring the extensibility point, with CreateSequenceResponse.		
Pass/Fail criteria		All step are as indicated.		
Notes		An attribute extensibility point is referred to using @{any} in place of the attribute name. This indicates that any attribute name can be used, from any namespace other than the wsrm: namespace.		

TP Id		TP/WAN/REC/WSI/RM/BV-002		
TP label		Consideration on the Use of "Piggy-Backing"		
Coverage	Spec	[OASIS WS-I RM]		
	Testable items	PiggyBack 1; O	PiggyBack 2; M	PiggyBack 3; R
Test purpose		Check that:		
		Some RM Protocol Header Blocks MAY be added to messages that are targeted to the same Endpoint to which those headers are to be sent (a concept often referred to as "piggy-backing"), thus saving the overhead of an additional message exchange.		
		[AND]		
		Reference parameters MUST be considered when determining whether two EPRs are targeted to the same Endpoint		
		[AND]		

	In order to ensure optimal and successful processing of RM Sequences, endpoints that receive RM-related messages SHOULD be prepared to process RM Protocol Header Blocks that are included in any message it receives.
Applicability	C_REC_000
Other PICS	
Initial condition	The simulated sender and the receiver under test are in the none sequence state.
Test procedure	1. The simulated ender sends a CreateSequence with an offer element.
	2. The receiver under test responds with CreateSequenceResponse.
	3. The sender sends a sequence message.
	4. The receiver responds including a SequenceAcknowledgement header block.
	❑ If the SOAP message also contains a sequence header block (piggy-backing), all the header blocks have the same endpoint reference (EPR).
	❑ If not, any other header block is sent in the same SOAP message, the receiver under test sends a message for every other RM-element (not piggy-backing).
	5. The sender responds using a SequenceAcknowledgement header block.
Pass/Fail criteria	In step 4, if the receiver sends only one message with more than one header block (piggy-backing), the endpoint reference (EPR) is the same for every header block.
Notes	See the sections of the WS-RM that define each RM Protocol header block for indications on which ones may be considered for piggy-backing.
	An endpoint reference is made using a "wsa:To" element. The way to test that every header block is targeted to the same endpoint is to check that there is only one "wsa:To" element in the soap:header.

TP Id		TP/WAN/REC/WSI/RM/BV-003		
TP label		Sequence Creation		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	WSAddress 1; C	SeqCreation 1; M	SeqCreation 3; M
		SeqCreation 6; M	SeqCreation 13; M	SeqCreation 16; M
		SeqCreation 17; M	SeqCreation 18; C	SeqCreation 19; M
		SeqCreation 20; O	SeqCreation 21; O	SeqCreation 23; M
		SeqRefused 1; M	Faults 3; M	
Test purpose		Check that:		
		When an Endpoint generates a message that carries an RM protocol element in the body of a SOAP envelope that Endpoint MUST include in that envelope a wsa:Action SOAP header block whose value is an IRI that is a concatenation of the WS-RM namespace URI, followed by a "/", followed by the value of the local name of the child element of the SOAP body.		
		[AND]		
		The RM Source MUST request creation of an outbound Sequence by sending a CreateSequence element in the body of a message to the RM Destination which in turn responds either with a message containing CreateSequenceResponse or a CreateSequenceRefused fault		
		[AND]		
		An offer is either accepted or rejected by the RM Destination in the CreateSequenceResponse message.		
		[AND]		
		The RM Destination MUST respond either with a CreateSequenceResponse response message or a CreateSequenceRefused fault		
		[AND]		
		An RM Destination MUST NOT accept (via the /wsrm:CreateSequenceResponse/wsrm:Accept element) an offer that contains the		

	"http://www.w3.org/2005/08/addressing/anonymous" IRI as its address
	[AND]
	The RM Destination MUST NOT send wsrm:CreateSequenceResponse element as a header block. This element is sent in the body of the response message in response to a CreateSequence request message
	[AND]
	The RM Destination MUST include wsrm:Identifier element within any CreateSequenceResponse message it sends. The RM Destination MUST set the value of this element to the absolute URI (conformant with RFC3986) that uniquely identifies the Sequence that has been created by the RM Destination
	[AND]
	wsrm:Expires element, if present, of type xs:duration accepts or refines the RM Source's requested duration for the Sequence. It specifies the amount of time after which any resources associated with the Sequence SHOULD be reclaimed thus causing the Sequence to be silently terminated
	[AND]
	The RM Destination MUST set the value of wsrm:Expires element to be equal to or less than the value requested by the RM Source in the corresponding CreateSequence message.
	[AND]
	wsrm:IncompleteSequenceBehaviour element, if present in wsrm:CreateSequenceResponse element, specifies the behavior that the destination will exhibit upon the closure or termination of an incomplete Sequence. For the purposes of defining the values used, the term "discard" refers to behavior equivalent to the Application Destination never processing a particular message.
	A value of "DiscardEntireSequence" indicates that the entire Sequence MUST be discarded if the Sequence is closed, or terminated, when there are one or more gaps in the final SequenceAcknowledgement.
	A value of "DiscardFollowingFirstGap" indicates that messages in the Sequence beyond the first gap MUST be discarded when there are one or more gaps in the final SequenceAcknowledgement.
	The default value of "NoDiscard" indicates that no acknowledged messages in the Sequence will be discarded.
	[AND]
	wsrm:Accept element, if present, enables an RM Destination to accept the offer if a corresponding Sequence for the reliable exchange of messages Transmitted from RM Destination to RM Source
	[AND]
	The RM Destination MUST include wsrm:AcksTo element within wsrm:Accept.
	[AND]
	CreateSequenceRefused properties:
	[Code] Sender or Receiver
	[Subcode] wsrm:CreateSequenceRefused
	[Reason] The Create Sequence request has been refused by the RM Destination.
	[Detail] xs:any
	[AND]
	Entities that generate WS-ReliableMessaging faults MUST include as the [action] property the default fault action IRI: http://docs.oasis-open.org/ws-rx/wsrm/200702/fault.
Applicability	C_REC_000
Other PICS	
Initial condition	The simulated sender and the receiver under test are in the none sequence state.
Test procedure	1. The simulated sender sends a CreateSequence, with an offer element message to the receiver under test.

		2. The receiver responds with a CreateSequenceResponse or a CreateSequenceRefused fault message.
		If the response is CreateSequenceResponse:
		3. The received message has the following properties:
		a. In the header block:
		❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/CreateSequenceResponse.
		❑ wsrm:CreateSequenceResponse is not present.
		b. In the body of the message:
		❑ the wsrm:Identifier value is an absolute URI that uniquely identifies the sequence created by the RM destination
		❑ the wsrm:Expires element, if present:
		• its type is xs:duration
		• is value is equal or less than the value requested by the RM source in the corresponding CreateSequence message
		• it is recommended that any resources associated with the sequence are reclaimed before this time. Otherwise, the sequence will be silently terminated after this time.
		❑ the wsrm:IncompleteSequenceBehaviour element may be present. Possible values are: "discard", "DiscardEntireSequence", "DiscardFollowingFirstGap" and "NoDiscard"
		❑ if the offer element contains the "http://www.w3.org/2005/08/addressing/anonymous" IRI as its address, the receiver does not accept this offer.
		❑ if wsrm:Accept is present, wsrm:AcksTo is present within the Accept element, and the receiver is able to send sequences messages
		❑ if wsrm:Accept is not present, the receiver is NOT able to send sequences messages.
		If the response is a CreateSequenceRefused fault:
		4. the received message has the following properties:
		❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/fault
		❑ Code = Sender or Receiver
		❑ Subcode = wsrm:CreateSequenceRefused
		❑ Reason = "The create sequence request has been refused by the RM destination".
		❑ Detail = xs:any.
Pass/Fail criteria		All the elements are as specified and only if the offer is accepted by the receiver, can it send sequence messages.
Notes		

TP Id		TP/WAN/REC/WSI/RM/BV-004		
TP label		Closing a Sequence		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	WSAddress 1; C	SeqClosing 1; O	SeqClosing 2; M
		SeqClosing 3; M	SeqClosing 4; R	SeqClosing 5; O
		SeqClosing 6; M	SeqClosing 8; O	SeqClosing 9; M
		SeqClosing 10; R	SeqClosing 11; M	SeqClosing 12; M
		SeqClosing 7; R		
Test purpose		Check that:		

When an Endpoint generates a message that carries an RM protocol element in the body of a SOAP envelope that Endpoint MUST include in that envelope a wsa:Action SOAP header block whose value is an IRI that is a concatenation of the WS-RM namespace URI, followed by a "/", followed by the value of the local name of the child element of the SOAP body.

[AND]

To ensure that the Sequence ends with a known final state either the RM Source or RM Destination MAY choose to close the Sequence before terminating it.

[AND]

If the RM Source wishes to close the Sequence, then it sends a CloseSequence element, in the body of a message, to the RM Destination. This message indicates that the RM Destination MUST NOT accept any new messages for the specified Sequence, other than those already accepted at the time the CloseSequence element is interpreted by the RM Destination

[AND]

Upon receipt of CloseSequence message, or subsequent to the RM Destination closing the Sequence of its own volition, the RM Destination MUST include a final SequenceAcknowledgement (within which the RM Destination MUST include the Final element) header block on any messages associated with the Sequence destined to the RM Source, including the CloseSequenceResponse message or on any Sequence fault Transmitted to the RM Source

[AND]

To allow the RM Destination to determine if it has received all of the messages in a Sequence, the RM Source SHOULD include the LastMsgNumber element in any CloseSequence messages it sends. The value of the LastMsgNumber element MUST be the same in all the CloseSequence messages for the closing Sequence

[AND]

If the RM Destination decides to close a Sequence of its own volition, it MAY inform the RM Source of this event by sending a CloseSequence element, in the body of a message, to the AcksTo EPR of that Sequence. The RM Destination MUST include a final SequenceAcknowledgement (within which the RM Destination MUST include the Final element) header block in this message and any subsequent messages associated with the Sequence destined to the RM Source

[AND]

While the RM Destination MUST NOT accept any new messages for the specified Sequence it MUST still process Sequence Lifecyle Messages and Acknowledgement Requests

[AND]

/wsrm:CloseSequence. This element MAY be sent by an RM Source to indicate that the RM Destination MUST NOT accept any new messages for this Sequence This element MAY also be sent by an RM Destination to indicate that it will not accept any new messages for this Sequence

[AND]

The RM Source or RM Destination MUST include wsrm:Identifier element in any CloseSequence messages it sends. The RM Source or RM Destination MUST set the value of this element to the absolute URI (conformant with RFC3986) of the closing Sequence

[AND]

The RM Source SHOULD include wsrm:LastMessageNumber element in any CloseSequence message it sends.

[AND]

A wsrm:CloseSequenceResponse element is sent in the body of a message in response to receipt of a CloseSequence request message. It indicates that the responder has closed the Sequence

[AND]

The responder (RM Source or RM Destination) MUST include wsrm:Identifier element in any CloseSequenceResponse messages it sends. The responder MUST set the value of this element to the absolute URI (conformant with RFC3986) of the closing Sequence.

[AND]

	In the case where the RM Destination wishes to discontinue use of a Sequence it is RECOMMENDED that it close the Sequence. Whenever possible the SequenceClosed fault SHOULD be used in place of the SequenceTerminated fault to allow the RM Source to still Receive Acknowledgements.
Applicability	C_REC_000
Other PICS	C_REC_WSI_033
Initial condition	The simulated sender has created a sequence with an offer element. The simulated sender and the receiver under test are in the created sequence state.

Test procedure	1. The simulated sender sends a sequence message including an AckRequested element in its header block or indicating that it is the last one.
	
	2. The receiver under test responds using a SequenceAcknowledgement header block.
	
	3. If C_REC_WSI_033 = TRUE, the receiver under test sends a CloseSequence element in the body of the message before the simulated sender does, check that the received message includes:
	
	    a. In the header block:
	
	        ❑ aSequenceAcknowledgement element and a wsrm:Final element within it are present.
	
	        ❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/CloseSequence.
	
	    b. In the body of the message, within the CloseSequence element:
	
	        ❑ the wsrm:Identifier value is an absolute URI of the closing sequence
	
	        ❑ it is recommended that a LastMsgNumber element is present and that in this case, it specifies the highest assigned message number of all the sequence traffic messages for the closing sequence.
	
	4. If the receiver under test sends the CloseSequence element in the body of the message before the simulated sender, the simulated sender responds with a CloseSequenceResponse message, including its Identifier element as an absolute URI, then goes to step 7 below.
	
	5. If C_REC_WSI_033 = FALSE, the simulated sender sends a CloseSequence element in the body of the message including a correct LastMsgNumber.
	
	6. The receiver responds with a message including:
	
	    a. In the header block:
	
	        ❑ A SequenceAcknowledgement header block including a Final element is present.
	
	        ❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/CloseSequenceResponse.
	
	    b. In the body of the message:
	
	        ❑ A CloseSequenceResponse element with a wsrm:Identifier element that is an absolute URI of the closing sequence.
	
	7. Once the sequence is closed, the sender sends a new sequence message referencing that closed sequence.
	
	8. The receiver under test does not accept that message. It is recommended that receiver responds with a SequenceClosed fault.

Pass/Fail criteria	All fields are as specified.
Notes	

TP Id	TP/WAN/REC/WSI/RM/BV-005
TP label	Sequence Termination

Coverage	Spec	[OASIS WS-I RM]		
	Testable items	WSAddress 1; C	SeqTermination 1; R	SeqTermination 2; M
		SeqTermination 3; O	SeqTermination 4; O	SeqTermination 5; M
		SeqTermination 7; M	SeqTermination 8; O	SeqTermination 9; M
		SeqTermination 10; M	SeqTermination 11; M	SeqTermination 12; R
		SeqTermination 13; M	SeqTermination 14; M	SeqTermination 15; M
Test purpose		Check that: When an Endpoint generates a message that carries an RM protocol element in the body of a SOAP envelope that Endpoint MUST include in that envelope a wsa:Action SOAP header block whose value is an IRI that is a concatenation of the WS-RM namespace URI, followed by a "/", followed by the value of the local name of the child element of the SOAP body. [AND] To allow the RM Destination to determine if it has received all of the messages in a Sequence, the RM Source SHOULD include the LastMsgNumber element in any TerminateSequence messages it sends [AND] The value of the LastMsgNumber element in the TerminateSequence message MUST be equal to the value of the LastMsgNumber element in any CloseSequence message(s) sent by the RM Source for the same Sequence [AND] If the RM Destination decides to terminate a Sequence of its own volition, it MAY inform the RM Source of this event by sending a TerminateSequence element, in the body of a message, to the AcksTo EPR for that Sequence. The RM Destination MUST include a final SequenceAcknowledgement (within which the RM Destination MUST include the Final element) header block in this message [AND] A wsrm:TerminateSequence element MAY be sent by an RM Source to indicate it has completed its use of the Sequence [AND] The RM Source MUST NOT send wsrm:TerminateSequence element as a header block [AND] Once wsrm:TerminateSequence element is sent, other than this element, the RM Source MUST NOT send any additional message to the RM Destination referencing this Sequence [AND] A wsrm:TerminateSequence element MAY also be sent by the RM Destination to indicate that it has unilaterally terminated the Sequence [AND] Upon sending wsrm:TerminateSequence message the RM Destination MUST NOT accept any additional messages (with the exception of the corresponding TerminateSequenceResponse) for this Sequence [AND] Upon receipt of a TerminateSequence the RM Source MUST NOT send any additional messages (with the exception of the corresponding TerminateSequenceResponse) for this Sequence [AND] The RM Source or RM Destination MUST include wsrm:Identifier element in any TerminateSequence message it sends. The RM Source or RM Destination MUST set the value of this element to the absolute URI (conformant with RFC3986) of the terminating Sequence [AND] /wsrm:TerminateSequence/wsrm:LastMsgNumber. The RM Source SHOULD include this element in any TerminateSequence message it sends. The LastMsgNumber element

	specifies the highest assigned message number of all the Sequence Traffic Messages for the terminating Sequence.
	[AND]
	TerminateSequenceResponse element is sent in the body of a message in response to receipt of a TerminateSequence request message. It indicates that the responder has terminated the Sequence. The responder MUST NOT send this element as a header block
	[AND]
	The responder (RM Source or RM Destination) MUST include wsrm:Identifier element in any TerminateSequenceResponse message it sends. The responder MUST set the value of this element to the absolute URI (conformant with RFC3986) of the terminating Sequence.
	[AND]
	On receipt of a TerminateSequence message the receiver (RM Source or RM Destination) MUST respond with a corresponding TerminateSequenceResponse message or generate a fault UnknownSequenceFault if the Sequence is not known.
Applicability	C_REC_000
Other PICS	C_REC_WSI_035
Initial condition	The simulated sender has created a sequence with an offer element. The simulated sender and the receiver under test are in the created sequence state.
Pass/Fail criteria	All fields and messages exchanged are as specified.
Test procedure	1. The simulated sender sends a sequence message including an AckRequested element in its header block or indicating that it is the last one.
	2. The receiver under test responds using a SequenceAcknowledgement header block.
	3. If C_REC_WSI_035 = TRUE, the receiver under test sends a TerminateSequence element in the body of the message before the simulated sender does so. The received message includes:
	a. In the header block:
	❑ A SequenceAcknowledgement element containing a wsrm:Final element.
	❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/TerminateSequence.
	❑ The wsrm: TerminateSequence is not present.
	b. In the body of the message, within the TerminateSequence element:
	❑ The wsrm:Identifier value is an absolute URI of the terminating sequence.
	❑ It is recommended that LastMsgNumber is present, and in that case, its value must be equal to the value of the LastMsgNumber element in any CloseSequence message(s) sent for the same sequence.
	4. If the receiver sends a TerminateSequence, the simulated sender responds with a TerminateSequenceResponse message, including its Identifier element as an absolute URI.
	5. If C_REC_WSI_035 = FALSE, the simulated sender sends a TerminateSequence element in the body of the message and it is recommended that the TerminateSequence element includes a correct LastMsgNumber.
	6. If the simulated sender has sent a TerminateSequence, the receiver generates an UnknownSequence fault or responds with a message including:
	a. In the header block:
	❑ A SequenceAcknowledgement header block.
	❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/TerminateSequenceResponse.
	b. In the body of the message within the TerminateSequenceResponse element:
	❑ wsrm:Identifier value = an absolute URI of the terminating sequence.
	7. Once the sequence is terminated, the simulated sender sends a sequence message referencing the terminated sequence.
	8. The receiver under test does not accept that message.

Notes	

TP Id		TP/WAN/REC/WSI/RM/BV-006		
TP label		Sequences		
Coverage	Spec	[OASIS WS-I RM]		
	Testable items	ProtocolInv 1; M	Sequences 1; M	Sequences 2; M
		Sequences 3; M	Sequences 4; M	Sequences 5; M
		Sequences 6; M	Sequences 7; M	Sequences 8; M
Test purpose		Check that: The RM Source MUST assign each message within a Sequence a message number beginning at 1 and increasing by exactly 1 for each subsequent message. These numbers MUST be assinged in the same order in which messages are sent by the Application Source. [AND] The RM Source MUST include a Sequence header block in all messages for which reliable transfer is REQUIRED [AND] The RM Source MUST identify Sequences with unique Identifier elements and the RM Source MUST assign each message within a Sequence a MessageNumber element that increments by 1 from an initial value of 1 [AND] The RM Source MUST NOT include more than one Sequence header block in any message [AND] The RM Destination MUST understand the Sequence header block [AND] The RM Source MUST assign a mustUnderstand attribute with a value 1/true (from the namespace corresponding to the version of SOAP to which the Sequence SOAP header block is bound) to the Sequence header block element. [AND] An RM Source that includes a Sequence header block in a SOAP envelope MUST include wsrm:Identifier element in that header block [AND] The RM Source MUST set the value of wsrm:Identifier element to the absolute URI (conformant with RFC3986) that uniquely identifies the Sequence [AND] The RM Source MUST include wsrm:MessageNumber element within any Sequence headers it creates. This element is of type MessageNumberType.			
Applicability		C_REC_000		
Other PICS				
Initial condition		The simulated sender has created a sequence with an offer element. The simulated sender and the receiver under test are in the created sequence state.			

Test procedure		1. The simulated sender sends a sequence message including an AckRequested element in its header block or indicates that it is the last one.
		
		2. The receiver under test responds using a SequenceAcknowledgement header block.
		
		3. If an offer element was sent in the CreateSequence and the receiver accepts that offer:
		   ❑ Wait until the receiver starts to send sequence messages.
		   ❑ In the received messages, check that:
		      • The wsrm:MessageNumber element is of type MessageNumberType and starts at 1 and increments by 1 for every sequential message.			

		• There is only one sequence header block in each message.
		• The wsrm:Identifier element must be present in the header block and must be an absolute URI that uniquely identifies the sequence.
		• The mustUnderstand attribute = "1" or "true".
Pass/Fail criteria		All elements in step 3 are as specified.
Notes		

TP Id		TP/WAN/REC/WSI/RM/BV-007		
TP label		Request Acknowledgement		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	ProtocolInv 6; R	WSAddress 3; C	ReqAck 1; O
		ReqAck 2; O	ReqAck 7; M	ReqAck 8; M
		SeqAck 3; R	SeqAck 4; M	SeqAck 21; R
		SeqAck 23; R		
Test purpose		Check that:		
		While the Sequence is not closed or terminated, the RM Source SHOULD retransmit unacknowledged messages.		
		[AND]		
		When an Endpoint generates an Acknowledgement Request that has no element content in the SOAP body, then the value of the wsa:Action IRI MUST be: http://docs.oasis-open.org/ws-rx/wsrm/200702/AckRequested.		
		[AND]		
		The RM Source MAY request an Acknowledgement Message from the RM Destination at any time by independently transmitting an AckRequested header block		
		[AND]		
		Alternatively the RM Source MAY include an AckRequested header block in any message targeted to the RM Destination		
		[AND]		
		An RM Source that includes an AckRequested header block in a SOAP envelope MUST include wsrm:Identifier element in that header block		
		[AND]		
		The RM Source MUST set the value of wsrm:Identifier element to the absolute URI, (conformant with RFC3986), that uniquely identifies the Sequence to which the request applies.		
		[AND]		
		The RM Source SHOULD process SequenceAcknowledgement header blocks that are included in any message it receives.		
		[AND]		
		If a non-mustUnderstand fault occurs when processing a SequenceAcknowledgement header that was piggy-backed, a fault MUST be generated, but the processing of the original message MUST NOT be affected		
		[AND]		
		Upon the receipt of a Nack, an RM Source SHOULD retransmit the message identified by the Nack.		
		[AND]		
		The RM Source SHOULD ignore a SequenceAcknowledgement containing a Nack for a message that has previously been acknowledged within an AcknowledgementRange.		
Applicability		C_REC_000 AND C_REC_WSI_036		
Other PICS				

Initial condition	The simulated sender and the receiver under test are in the none sequence state.
Test procedure	1. The simulated sender sends a CreateSequence message with an offer element.
	If the receiver accepts the offer:
	2. The receiver responds with a CreateSequenceResponse including an accept element.
	3. The simulated sender sends a sequence message indicating that it is the last one.
	4. The receiver under test sends a SequenceAcknowledgement and starts to send sequence messages and sends its first AckRequested element in the header block of one message.
	5. In that received message in the header block, check that:
	❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/AckRequested (if soap body is empty).
	❑ wsrm:Identifier = absolute URI that uniquely identifies the sequence.
	6. The simulated sender does not validate any message with a SequenceAcknowledgement header block with a None element.
	7. The receiver should retransmit the messages.
	8. If the receiver retransmits the messages, the simulated sender does not validate any message using a Nack element within a SequenceAcknowledgement header block.
	9. The receiver should retransmit the messages.
	10. If the receiver retransmits the messages, the simulated sender validates the messages using a SequenceAcknowledgement header block.
	11. The simulated sender sends a Nack element with the MessageNumber of one of the previous messages received.
	12. The receiver should ignore the Nack element.
Pass/Fail criteria	All elements are as specified.
	If a non-mustUnderstand fault occurs when processing a SequenceAcknowledgement header that was piggy-backed, a fault is generated, but the processing of the original message is not affected.
Notes	

TP Id		TP/WAN/REC/WSI/RM/BV-008		
TP label		Sequence Acknowledgement		
Coverage	Spec	[OASIS WS-I RM]		
	Testable items	ProtocolInv 2; M	ProtocolInv 3; M	ProtocolInv 4; M
		ProtocolInv 5; O	WSAddress 2; C	ReqAck 3; R
		ReqAck 4; M	ReqAck 5; M	SeqAck 1; O
		SeqAck 2; O	SeqAck 5; O	SeqAck 6; M
		SeqAck 7; R	SeqAck 8; M	SeqAck 9; M
		SeqAck 10; M	SeqAck 11; O	SeqAck 12; M
		SeqAck 13; M	SeqAck 14; M	SeqAck 15; M
		SeqAck 16; M	SeqAck 17; M	SeqAck 18; O
		SeqAck 19; M	SeqAck 20; O	SeqAck 22; M
Test purpose		Check that:		
		Within every Acknowledgement Message it issues, the RM Destination MUST include one or more AcknowledgementRange child elements that contain, in their collective ranges, the message number of every message accepted by the RM Destination.		
		[AND]		
		The RM Destination MUST exclude, in the AcknowledgementRange elements, the message numbers of any messages it has not accepted.		

	[AND]
	If no messages have been received the RM Destination MUST return None instead of an AcknowledgementRange(s).
	[AND]
	The RM Destination MAY transmit a Nack for a specific message or messages instead of an AcknowledgementRange(s).
	[AND]
	When an Endpoint generates an Acknowledgement Message that has no element content in the SOAP body, then the value of the wsa:Action IRI MUST be: http://docs.oasis-open.org/ws-rx/wsrm/200702/SequenceAcknowledgement
	[AND]
	The RM Destination SHOULD process AckRequested header blocks that are included in any message it receives
	[AND]
	If a non-mustUnderstand fault occurs when processing an AckRequested header block that was piggy-backed, a fault MUST be generated, but the processing of the original message MUST NOT be affected.
	[AND]
	An RM Destination that Receives a message that contains an AckRequested header block MUST send a message containing a SequenceAcknowledgement header block to the AcksTo endpoint reference for a known Sequence or else generate an UnknownSequence fault
	[AND]
	The RM Destination MAY Transmit the SequenceAcknowledgement header block independently (i.e. as a header of a SOAP envelope with an empty body)
	[AND]
	Alternatively, an RM Destination MAY include a SequenceAcknowledgement header block on any SOAP envelope targeted to the endpoint referenced by the AcksTo EPR
	[AND]
	During creation of a Sequence the RM Source MAY specify the WS-Addressing anonymous IRI as the address of the AcksTo EPR for that Sequence
	[AND]
	When the RM Source specifies the WS-Addressing anonymous IRI as the address of the AcksTo EPR, the RM Destination MUST Transmit any SequenceAcknowledgement headers for the created Sequence in a SOAP envelope to be Transmitted on the protocol binding-specific back-channel
	[AND]
	When the RM Destination receives an AckRequested header, and the AckTo EPR for that sequence is the WS-Addressing anonymous IRI, the RM Destination SHOULD respond on the protocol binding-specific back-channel provided by the Received message containing the AckRequested header block
	[AND]
	An RM Destination that includes a SequenceAcknowledgement header block in a SOAP envelope MUST include wsrm:Identifier element in that header block
	[AND]
	The RM Destination MUST set the value of wsrm:Identifier element to the absolute URI (conformant with RFC3986) that uniquely identifies the Sequence
	[AND]
	The RM Destination MUST NOT include multiple SequenceAcknowledgement header blocks that share the same value for Identifier within the same SOAP envelope
	[AND]
	The RM Destination MAY include one or more instances of wsrm:AcknowledgementRange element within a SequenceAcknowledgement header block

	[AND]
	The ranges in wsrm:AcknoledgementRange MUST NOT overlap.
	[AND]
	The RM Destination MUST NOT include wsrm:AcknowledgementRange element if a sibling Nack or None element is also present as a child of SequenceAcknowledgement
	[AND]
	The RM Destination MUST set the value of upper attribute (in wsrm:AcknoledgementRange) equal to the message number of the highest contiguous message in a Sequence range accepted by the RM Destination
	[AND]
	The RM Destination MUST set the value of Lower attribute (in wsrm:AcknowledgementRange) equal to the message number of the lowest contiguous message in a Sequence range accepted by the RM Destination
	[AND]
	The RM Destination MUST include wsrm:None element within a SequenceAcknowledgement header block if the RM Destination has not accepted any messages for the specified Sequence
	[AND]
	The RM Destination MUST NOT include wsrm:None element if a sibling AcknowledgementRange or Nack element is also present as a child of the SequenceAcknowledgement
	[AND]
	The RM Destination MAY include wsrm:Final element within a SequenceAcknowledgement header block. This element indicates that the RM Destination is not receiving new messages for the specified Sequence
	[AND]
	The RM Destination MUST include wsrm:Final element when the Sequence is closed. The RM Destination MUST NOT include this element when sending a Nack; it can only be used when sending AcknowledgementRange elements or a None
	[AND]
	The RM Destination MAY include wsrm:Nack element within a SequenceAcknowledgement header block. If used, the RM Destination MUST set the value of this element to a MessageNumberType representing the MessageNumber of an unreceived message in a Sequence. The RM Destination MUST NOT include a Nack element if a sibling AcknowledgementRange or None element is also present as a child of SequenceAcknowledgement
	[AND]
	The RM Destination MUST NOT issue a SequenceAcknowledgement containing a Nack for a message that it has previously acknowledged within an AcknowledgementRange
Applicability	C_REC_000
Other PICS	
Initial condition	The simulated sender has created a sequence with an offer. The simulated sender and the receiver under test are in the created sequence state.
Test procedure	1. The simulated sender transmits 3 messages with its respective sequence header block and in the last one it includes an AckRequest. 2. The receiver under test responds including a SequenceAcknowledgement header block or an UnknownSequence fault. 3. If the response has a SequenceAcknowledgement header block: ❑ If AcksTo field of any message to be acknowledged is an anonymous IRI, the receiver must transmit the SequenceAcknowledgement on the channel provided by the context of the received message containing a SOAP envelope that contains a sequence header block and/or an AckRequested header block for that same sequence identifier.

		❑ If the soap body of the message is empty, the wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/SequenceAcknowledgement
		❑ The wsrm:Identifier = absolute URI. It cannot be used in another SequenceAcknowledgement in the same message.
		❑ Only one of these elements is present: one or more AcknowledgementRange, a None or a Nack.
		❑ The final element is present when the sequence is closed, but it is not included when a Nack is sent.
		❑ If an AcknowledgementRange element is present:
		• the lower attribute is equal to or less than the upper attribute,
		• the lower attribute is equal to the message number of the lowest contiguous message in a sequence range accepted by the receiver,
		• the upper attribute is equal to the message number of the highest contiguous message in a sequence range accepted by the receiver.
		❑ If a None element is present then no messages have been accepted or received.
		❑ If a Nack element is present a specific message has not been received and it cannot be included in a SequenceAcknowledgement header block for a message that it had previously acknowledged within an AcknowledgementRange.
Pass/Fail criteria		All elements are as specified. If a non-mustUnderstand fault occurs when processing an AckRequested header block that was piggy-backed, a fault is generated, but the processing of the original message is not affected.
Notes		

TP Id		TP/WAN/REC/WSI/RM/BV-009
TP label		Sequence Terminated Fault
Coverage	**Spec**	[OASIS WS-I RM]
	Testable items	SeqTerminatedFault 2; M SeqTerminatedFault 3; M SeqTerminatedFault 4; M
		Faults 1; R Faults 2; M Faults 3; M
Test purpose		Check that: SequenceTerminated has the following properties: [Code] Sender or Receiver [Subcode] wsrm:SequenceTerminated [Reason] The Sequence has been terminated due to an unrecoverable error. [Detail] <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier> [AND] An Endpoint MUST generate a SequenceTerminated fault when encountering an unrecoverable condition or detection of violation of the protocol [AND] An Endpoint that receives a SequenceTerminte fault MUST terminate the Sequence if not otherwise terminated [AND] Destinations that generate faults related to known sequences SHOULD transmit those faults. [AND] If transmitted, faults MUST be transmitted to the same [destination] as Acknowledgement messages [AND]

	Entities that generate WS-ReliableMessaging faults MUST include as the [action] property the default fault action IRI: http://docs.oasis-open.org/ws-rx/wsrm/200702/fault.
Applicability	C_REC_000
Other PICS	C_REC_WSI_035
Initial condition	The simulated sender and the receiver under test are in the none sequence state. The simulated sender is able to send a sequence message when the sequence has been terminated.
Test procedure	1. The simulated sender sends a CreateSequence message with an offer element.
	2. The receiver under test responds with a CreateSequenceResponse message accepting the offer.
	3. The simulated sender sends a sequence indicating that it is the last message.
	4. The receiver responds with a SequenceAcknowledgement with the element AcknowledgementRange Lower=1 and Upper=1.
	5. IF C_REC_WSI_035=TRUE, wait until the receiver under test sends a TerminateSequence or force it to terminate the sequence and the simulated sender responds with TerminateSequenceResponse. ELSE, the simulated sender sends a TerminateSequence message and the receiver under test responds with TerminateSequenceResponse.
	6. The simulated sender transmits a sequence with the message number within the range, for example, Message Number=2.
	7. The receiver generates a SequenceTerminated fault. It is recommended that the fault is transmitted to the sender.
	8. If the fault is transmitted by the receiver under test, the message includes the following properties:
	❏ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/fault.
	❏ Code = Sender
	❏ Subcode = SequenceTerminated
	❏ Reason = The Sequence has been terminated due to an unrecoverable error.
	❏ Detail = <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
	9. The simulated sender terminates the sequence and passes to the none sequence state.
Pass/Fail criteria	All elements are as specified in step 8
Notes	

TP Id		TP/WAN/REC/WSI/RM/BV-010		
TP label		Unknown Sequence Fault		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	UnknownSeq 1; M	UnknownSeq 2; M	UnknownSeq 3; M
		Faults 1; R	Faults 2; M	Faults 3; M
Test purpose		Check that:		
		UnknownSequence has the following properties:		
		[Code] Sender		
		[Subcode] wsrm:UnknownSequence		
		[Reason] The value of wsrm:Identifier is not a known Sequence identifier		
		[Detail] <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>		
		[AND]		
		An Endpoint MUST generate an UnknownSequence fault in response to a message containing an unknown or terminated Sequence identifier		
		[AND]		

		An Endpoint that receives an UnknownSequence fault MUST terminate the Sequence if not otherwise terminated
		[AND]
		Destinations that generate faults related to known sequences SHOULD transmit those faults.
		[AND]
		If transmitted, faults MUST be transmitted to the same [destination] as Acknowledgement messages
		[AND]
		Entities that generate WS-ReliableMessaging faults MUST include as the [action] property the default fault action IRI: http://docs.oasis-open.org/ws-rx/wsrm/200702/fault.
Applicability		C_REC_000
Other PICS		
Initial condition		The simulated sender and the receiver under test are in the none sequence state. The simulated sender is able to send a sequence message when it is in the none sequence state.
Test procedure		1. The simulated sender transmits a sequence message to the receiver under test.
		2. The receiver generates an UnknownSequence fault. It is recommended that the fault is transmitted to the sender.
		3. If the fault is transmitted by the receiver under test, that message includes the following properties:
		❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/fault.
		❑ Code = Sender
		❑ Subcode = UnknownSequence
		❑ Reason = The value of wsrm:Identifier is not a known sequence identifier
		❑ Detail = <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
		4. Wait until the receiver terminates the sequence.
Pass/Fail criteria		All elements are as specified in step 3
Notes		

TP Id		TP/WAN/REC/WSI/RM/BV-011		
TP label		Invalid Acknowledgement Fault		
Coverage	Spec	[OASIS WS-I RM]		
	Testable items	InvalidAck 1; M	InvalidAck 2; M	Faults 1; R
		Faults 2; M	Faults 3; M	
Test purpose		Check that:		
		InvalidAcknowledgement fault has the following properties:		
		[Code] Sender		
		[Subcode] wsrm:InvalidAcknowledgement		
		[Reason] The SequenceAcknowledgement violates the cumulative Acknowledgement invariant.		
		[Detail] <wsrm:SequenceAcknowledgement ...> ... </wsrm:SequenceAcknowledgement>		
		[AND]		
		RM Source MUST generate an InvalidAcknowledgement in response to a SequenceAcknowledgement that violate the invariants stated in 2.3 or any of the requirements in 3.9 about valid combinations of AckRange, Nack and None in a single SequenceAcknowledgement element or with respect to already Received such elements.		
		[AND]		
		Destinations that generate faults related to known sequences SHOULD transmit those faults.		

	[AND] If transmitted, faults MUST be transmitted to the same [destination] as Acknowledgement messages [AND] Entities that generate WS-ReliableMessaging faults MUST include as the [action] property the default fault action IRI: http://docs.oasis-open.org/ws-rx/wsrm/200702/fault.
Applicability	C_REC_000
Other PICS	
Initial condition	The simulated sender and the receiver under test are in the none sequence state. Simulated Sender is able to include a wrong AckRange and None and Nack elements in a SequenceAcknowledgement message.

Test procedure	1. The simulated sender creates a sequence with an offer.
	
	2. The receiver under test responds using a CreateSequenceResponse message accepting the offer.
	
	3. After the simulated sender has sent its sequences and the receiver acknowledges them, the receiver under test sends a sequence message with its respective message number.
	
	4. If the last sequence message is not labelled as the last one, wait until the receiver sends an AckRequested. Otherwise, go to next step.
	
	5. The simulated sender responds with a SequenceAcknowledgement with the wrong AckRange element and None and Nack elements.
	
	6. The receiver generates an InvalidAcknowledgement fault. It is recommended that the fault is transmitted to the sender.
	
	7. If the fault is transmitted by the receiver under test, that message includes the following properties:
	
	   ❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/fault.
	
	   ❑ Code = Sender
	
	   ❑ Subcode = InvalidAcknowledgement
	
	   ❑ Reason = <any>
	
	   ❑ Detail = <any related to the message that produces the fault>

Pass/Fail criteria	All elements are as specified in step 7.
Notes	

TP Id		TP/WAN/REC/WSI/RM/BV-012
TP label		Message Number Rollover
Coverage	**Spec**	[OASIS WS-I RM]
	Testable items	MessageNumrRoll 4; R		
Test purpose		Check that: RM Source SHOULD continue to retransmit undelivered messages until the Sequence is closed or terminated.
Applicability		C_REC_000
Other PICS		
Initial condition		The simulated sender has created a sequence with an offer element. The simulated sender and the receiver under test are in the created sequence state. The simulated sender is able to send a message number rollover fault instead of a SequenceAcknowledgement message.
Test procedure		1. The simulated sender under test transmits a sequence message indicating that it is the last one. 2. The receiver under test sends its sequence and the SequenceAcknowledgement. 3. The simulated sender generates a message number rollover fault and this is transmitted

	to the receiver.	
	4. The receiver should retransmit undelivered messages until the sender closes or terminates the sequence.	
	5. The simulated sender closes the sequence.	
Pass/Fail criteria	Step 4 must be as indicated.	
Notes		

TP Id		TP/WAN/REC/WSI/RM/BV-012_B		
TP label		Message Number Rollover2		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	MessageNumrRoll 1; M	MessageNumrRoll 2; M	MessageNumrRoll 3; R
		Faults 1; R	Faults 2; M	Faults 3; M
Test purpose		Check that: Message Number Rollover fault has the following properties: [Code] Sender [Subcode] wsrm:MessageNumberRollover [Reason] The maximum value for wsrm:MessageNumber has been exceeded. [Detail] <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier> <wsrm:MaxMessageNumber> wsrm:MessageNumberType </wsrm:MaxMessageNumber> [AND] RM Destination MUST generate a MessageNumberRollover fault when Message number in /wsrm:Sequence/wsrm:MessageNumber of a Received message exceeds the internal limitations of an RM Destination or reaches the maximum value of 9,223,372,036,854,775,807 [AND] RM Destination SHOULD continue to accept undelivered messages until the Sequence is closed or terminated [AND] Destinations that generate faults related to known sequences SHOULD transmit those faults. [AND] If transmitted, faults MUST be transmitted to the same [destination] as Acknowledgement messages [AND] Entities that generate WS-ReliableMessaging faults MUST include as the [action] property the default fault action IRI: http://docs.oasis-open.org/ws-rx/wsrm/200702/fault.		
Applicability		C_REC_000		
Other PICS				
Initial condition		The simulated sender has created a sequence with an offer element. The simulated sender and the receiver under test are in the created sequence state. The simulated sender is able to change the message number of its sequence message.		
Test procedure		1. The simulated sender sends a sequence message with a MessageNumber=1 2. The receiver under test responds with its sequence message and may include a SequenceAcknowledge header block. 3. The simulated sender transmits a sequence message with a message number outside the range (bigger than 9,223,372,036,854,775,807 or its internal limitation). 4. The receiver generates a message number rollover fault. It is recommended that the fault is transmitted to the sender. 5. If the fault is transmitted by the receiver under test, that message includes the following properties:		

	❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/fault.
	❑ Code = Sender
	❑ Subcode = MessageNumberRollover
	❑ Reason = The maximum value for wsrm:MessageNumber has been exceeded
	❑ Detail = <wsrm:Identifier ...> *xs:anyURI* </wsrm:Identifier>
	<wsrm:MaxMessageNumber> *wsrm:MessageNumberType* </wsrm:MaxMessageNumber>
	6. The simulated sender retransmits its undelivered messages.
	7. The receiver should accept undelivered messages until the sequence is closed or terminated.
	8. The simulated sender closes the sequence.
Pass/Fail criteria	All elements are as specified in step 5 and steps 2, 4 and 7 must be as indicated.
Notes	

TP Id		TP/WAN/REC/WSI/RM/BV-013		
TP label		Sequence Closed Fault		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	SeqClosedFault 1; M	SeqClosedFault 2; M	Faults 1; R
		Faults 2; M	Faults 3; M	
Test purpose		Check that: SequenceClosed fault MUST be generated by an RM Destination when it is asked to accept a message for a Sequence that is closed [AND] SequenceClosed properties: [Code] Sender [Subcode] wsrm:SequenceClosed [Reason] The Sequence is closed and cannot accept new messages. [Detail] <wsrm:Identifier...> xs:anyURI </wsrm:Identifier> [AND] Destinations that generate faults related to known sequences SHOULD transmit those faults. [AND] If transmitted, faults MUST be transmitted to the same [destination] as Acknowledgement messages [AND] Entities that generate WS-ReliableMessaging faults MUST include as the [action] property the default fault action IRI: http://docs.oasis-open.org/ws-rx/wsrm/200702/fault.			
Applicability		C_REC_000		
Other PICS				
Initial condition		The simulated sender has created a sequence with an offer element. The simulated sender and receiver under test are in the created sequence state. The simulated sender is able to send a sequence message as long as the sequence has not yet been closed.			
Test procedure		1. The simulated sender sends a sequence to the receiver under test with MessageNumber=1 and indicating that it is the last one. 2. The receiver responds with a SequenceAcknowledgement with an AcknowledgementRange Lower=1 Upper=1, and a sequence message 3. The simulated sender sends a CloseSequence. 4. The receiver responds with CloseSequenceResponse.			

	5.	The simulated sender transmits a sequence with a message number within the range, for example, MessageNumber=2.
	6.	The receiver generates a SequenceClosed fault. It is recommended that the fault is transmitted to the sender.
	7.	If the fault is transmitted by the receiver under test, that message includes the following properties:

❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/fault.

❑ Code = Sender

❑ Subcode = SequenceClosed

❑ Reason = The Sequence is closed and cannot accept new messages

❑ Detail = <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>

| **Pass/Fail criteria** | All elements are as specified in step 7. |
| **Notes** | |

TP Id		TP/WAN/REC/WSI/RM/BV-014		
TP label		WSRM Required Fault		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	WSRMReq 1; C	WSRMReq 2; M	Faults 1; R
		Faults 2; M	Faults 3; M	
Test purpose		Check that:		

Check that:

If an RM Destination requires the use of WS-RM, WSRMRequired fault is generated when it Receives an incoming message that did not use this protocol.

[AND]

WSRM Required properties:

[Code] Sender

[Subcode] wsrm:WSRMRequired

[Reason] The RM Destination requires the use of WSRM

[Detail] xs:any

[AND]

Destinations that generate faults related to known sequences SHOULD transmit those faults.

[AND]

If transmitted, faults MUST be transmitted to the same [destination] as Acknowledgement messages

[AND]

Entities that generate WS-ReliableMessaging faults MUST include as the [action] property the default fault action IRI: http://docs.oasis-open.org/ws-rx/wsrm/200702/fault.

Applicability	C_REC_000 AND C_REC_WSI_034
Other PICS	
Initial condition	The simulated sender and the receiver under test are in the none sequence state. The simulated sender is able to send a message without using WSRM protocol.
Test procedure	1. The simulated sender transmits a SOAP message without using any element of the WSRM protocol.
	2. The receiver generates a WSRMRequired fault. It is recommended that the fault is transmitted to the sender.
	3. If the fault is transmitted by the receiver under test, that message includes the following properties:

❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/fault.

	❑ Code = Sender
	❑ Subcode = WSRMRequired
	❑ Reason = The RM Destination requires the use of WSRM
	❑ Detail = xs:any
Pass/Fail criteria	All elements are as specified in step 3.
Notes	

Bibliography

[b-ITU-T H.810 (2013)]   Recommendation ITU-T H.810 (2013), *Interoperability design guidelines for personal health systems*.

[b-CDG 1.0]   Continua Health Alliance, Continua Design Guidelines v1.0. (2008), *Continua Design Guidelines*.

[b-CDG 2010]   Continua Health Alliance, Continua Design Guidelines v1.5 (2010), *Continua Design Guidelines*.

[b-CDG 2011]   Continua Health Alliance, Continua Design Guidelines (2011), "Adrenaline", *Continua Design Guidelines*.

[b-CDG 2012]   Continua Health Alliance, Continua Design Guidelines (2012), "Catalyst", *Continua Design Guidelines.*

[b-CDG 2013]   Continua Health Alliance, Continua Design Guidelines (2013), "Endorphin" *Continua Design Guidelines.*

[b-ETSI SR 001 262]   ETSI SR 001 262 v1.8.1 (2003), *ETSI drafting rules*.
https://docbox.etsi.org/MTS/MTS/10-PromotionalMaterial/MBS-20111118/Referenced%20Documents/Drafting%20Rules.pdf

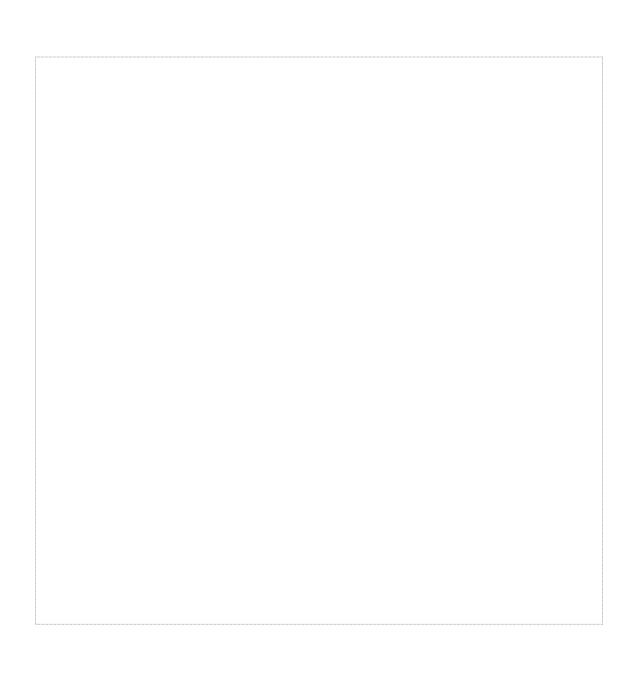[b-IEEE 11073-20601-2008] IEEE 11073-20601: 2008, *Health informatics – Personal health device communication – Part 20601: Application profile – Optimized exchange protocol.*
https://standards.ieee.org/findstds/standard/11073-20601-2008.html

[b-SOAP 1.2]   W3C SOAP 1.2 (2007), *SOAP Version 1.2* (Second Edition).
http://www.w3.org/TR/soap/

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems