International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.831

(01/2015)

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 1: Web services interoperability: Sender

Recommendation ITU-T H.831

International Telecommunication Union

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	H.230–H.239
Communication procedures	H.240–H.259
Coding of moving video	H.260–H.279
Related systems aspects	H.280–H.299
Systems and terminal equipment for audiovisual services	H.300–H.349
Directory services architecture for audiovisual and multimedia services	H.350–H.359
Quality of service architecture for audiovisual and multimedia services	H.360–H.369
Telepresence	H.420–H.429
Supplementary services for multimedia	H.450–H.499
MOBILITY AND COLLABORATION PROCEDURES	
Overview of Mobility and Collaboration, definitions, protocols and procedures	H.500–H.509
Mobility for H-Series multimedia systems and services	H.510–H.519
Mobile multimedia collaboration applications and services	H.520–H.529
Security for mobile multimedia systems and services	H.530–H.539
Security for mobile multimedia collaboration applications and services	H.540–H.549
Mobility interworking procedures	H.550–H.559
Mobile multimedia collaboration inter-working procedures	H.560–H.569
BROADBAND, TRIPLE-PLAY AND ADVANCED MULTIMEDIA SERVICES	
Broadband multimedia services over VDSL	H.610–H.619
Advanced multimedia services and applications	H.620–H.629
Ubiquitous sensor network applications and Internet of Things	H.640–H.649
IPTV MULTIMEDIA SERVICES AND APPLICATIONS FOR IPTV	
General aspects	H.700–H.719
IPTV terminal devices	H.720–H.729
IPTV middleware	H.730–H.739
IPTV application event handling	H.740–H.749
IPTV metadata	H.750–H.759
IPTV multimedia application frameworks	H.760–H.769
IPTV service discovery up to consumption	H.770–H.779
Digital Signage	H.780–H.789
E-HEALTH MULTIMEDIA SERVICES AND APPLICATIONS	
Interoperability compliance testing of personal health systems (HRN, PAN, LAN, TAN and WAN)	**H.820–H.859**
Multimedia e-health data exchange services	H.860–H.869

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T H.831

Conformance of ITU-T H.810 personal health devices: WAN interface Part 1: Web services interoperability: Sender

Summary

Recommendation ITU-T H.831 is a transposition of Continua Test Tool DG2013, Test Suite Structure & Test Purposes, WAN Interface; Part 1: Web Services Interoperability. Sender (Version 1.4, 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.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T H.831	2015-01-13	16	11.1002/1000/12249

* 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/11.1002/1000/11830-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: 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 Test Tool DG2013, Test Suite Structure & Test Purposes, WAN Interface; Part 1: Web Services Interoperability. Sender (Version 1.4, 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
1.2	2012-10-05	Initial release for Test Tool DG2011. It is the same version as "TSS&TP_1.5_WAN_PART_1_(SEN WS-I)_v1.2.doc" because new features included in [b-CDG 2011] do not affect the test procedures specified in this document.
1.3	2013-05-24	Initial release for Test Tool DG2012. It is the same version as "TSS&TP_DG2011_WAN_PART_1_(SEN WS-I)_v1.2.doc" because new features included in [b-CDG 2012] do not affect the test procedures specified in this document.
1.4	2014-01-24	Initial release for Test Tool DG2013. It is the same version as "TSS&TP_DG2012_WAN_PART_1_(SEN WS-I)_v1.2.doc" because new features included in CDG 2013 [ITU-T H.810] do not affect the test procedures specified in this document.

Recommendation ITU-T H.831

Conformance of ITU-T H.810 personal health devices: WAN interface Part 1: Web services interoperability: Sender

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 Continua specifications. 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 document has been divided into a set of eight parts. Each part contains:

– Part 1: Web Services Interoperability. Sender
– Part 2: Web Services Interoperability. 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] Recommendation ITU-T H.810 (2013), *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 Security Profile Version 1.1*.
<http://www.ws-i.org/Profiles/BasicProfile-1.1.html>

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

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

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

3 Definitions

3.1 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
ATS	Abstract Test Suite
ATNA	Audit Trail and Node Authentication
CDG	Continua Design Guidelines
DUT	Device Under Test
EPR	Endpoint Reference
GUI	Graphical User Interface
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 Health Manager
PICS	Protocol Implementation Conformance Statement
PIXIT	Protocol Implementation extra Information for Testing
SDP	Service Discovery Protocol
SOAP	Simple Object Access Protocol
STR	Security Token Reference
TCRL	Test Case Reference List
TCWG	Test and Certification Working Group
TP	Test Purposes
URI	Uniform Resource Identifier

TSS Test Suite Structure

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 document 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 to the various versions of the CDG

CDG name	Transposed as	Version	Description	Designation
2013 plus errata	ITU-T H.810	4.1	CDG 2013 plus errata noting all ratified bugs.	–
2013	–	4.0	Release 2013 of CDG including maintenance updates of the CDG 2012 and additional guidelines that cover new functionalities.	Endorphin
2012 plus errata	–	3.1	CDG 2012 plus errata noting all ratified bugs [b-CDG 2012].	–
2012	–	3.0	Release 2012 of the CDG including maintenance updates of 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 CDG including maintenance updates of CDG 2010 and additional guidelines that cover new functionalities [b-CDG 2011].	Adrenaline

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

CDG name	Transposed as	Version	Description	Designation
2010 plus errata	–	1.6	CDG 2010 integrated with identified errata	–
2010	–	1.5	Release 2010 of CDG with maintenance updates of 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 group 1.1 (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 analyser (BCA)
    - Subgroup 1.4.14: Basic electrocardiograph (ECG)

- 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 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)
- Group 2.5: Consent management (CM)
  - Subgroup 2.5.1: WAN XDR transaction (TRANS)
  - Subgroup 2.5.2: WAN service validation (SER)

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 (TP)

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

A.1 TP definition conventions

The test purposes 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 label**: This is the TP's title.
- **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 were 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.).
- **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 1.1.1 – Basic profile (BP)

TP Id		TP/WAN/SEN/WSI/BP/BV-000		
TP label		SOAP Envelope 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-R1032; M		
Applicability		C_SEN_000		
Initial condition		The simulated receiver has a WebService enabled with many different services and the sender under test has a SOAP message ready to be sent to the respective service according to its needs.		
Test procedure		1. The sender under test sends the SOAP message to the receiver. 2. Check that the captured message has the following structure: <soap:Envelope *'namespace'*> <soap:Header> ... </soap:Header> <soap:Body> *Here are the children of soap:envelope* </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 has, in this order, an envelope, an optional header and a body. • The namespaces that appear in the soap message are qualified. • 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. • The SOAP envelope's namespace is "http://www.w3.org/2003/05/soap-envelope" to support SOAP 1.2 [b-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 (not compliant with the WS-I Profile).		

TP Id		TP/WAN/SEN/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
Applicability		C_SEN_000		
Initial condition		The simulated receiver has a WebService enabled with many different services and the sender under test has a SOAP message ready to be sent to the respective service according to its needs.		
Test procedure		1. Make the sender under test send a SOAP message. 2. Check within the captured message that: a. the soap:encodingStyle attribute is present and the envelope contains: ❑ a namespace which is not "http://schemas.xmlsoap.org/soap/envelope/" ❑ an element that is not a child of soap:Body. ❑ If an rpc-literal binding is used, check that the element is not a grandchild of soap:body.		
Pass/Fail criteria		If present, the soap:encodingStyle attribute is as specified within the test procedure above.		
Notes				

TP Id		TP/WAN/SEN/WSI/BP/BV-002		
TP label		Use of SOAP in HTTP		
Coverage	**Spec**	[OASIS/WS-I BP]		
	Testable items	BP-R1132; M	BP-R1140; M	
Applicability		C_SEN_000		
Initial condition		The simulated receiver has a WebService enabled and the sender under test is ready to send an HTTP request.		
Test procedure		1. Make the sender under test send a message to the simulated receiver using the HTTP protocol. 2. Check in the HTTP header of the captured message that: a. the HTTP version is 1.1 b. POST method is used.		
Pass/Fail criteria		Check that all values are as specified in the HTTP header.		
Notes				

TP Id		TP/WAN/SEN/WSI/BP/BV-003		
TP label		HTTP Status Codes		
Coverage	**Spec**	[OASIS/WS-I BP]		
	Testable items	BP-R1131; O		
Applicability		C_SEN_000		
Initial condition		The simulated receiver has a WebService enabled with many different services and the sender under test has an HTTP request ready to be sent to the respective service according to its needs.		
Test procedure		1. Make the sender under test send an HTTP request to the receiver. 2. The simulated receiver responds with "307 Temporary Redirect" as the status code. 3. If C_SEN_WSI_001=TRUE, the sender redirects the request, or else the sender does not redirect the request.		
Pass/Fail criteria		If C_SEN_WSI_001=TRUE, the sender redirects the request to the http address indicated in the "307 Temporary Redirect" HTTP response.		
Notes				

TP Id		TP/WAN/SEN/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		
Applicability		C_SEN_000		
Initial condition		The simulated receiver has a WebService enabled with many different services and the sender under test has a SOAP message ready to be sent to the respective service according to its needs.		
Test procedure		1. Wait until the sender under test sends a SOAP message or, if necessary, force it to send a SOAP message. 2. Take the WSDL description of the web service using its URL and check the soap envelope of the captured message: *If an rpc-literal binding is used:* a. If the soapbind:body element of the description is an empty string, there is no part accessor elements. b. If the soapbind:body element of the description is not empty, check that the part accessor of the envelope is present and that there is no xsi:nil attribute with a value of "1" or "true". *If doc-literal binding is used:* a. If the value of the parts attribute of soapbind:body is an empty string, the envelope does not have element content in soap:Body element.		
Pass/Fail criteria		Check that the envelope is as specified in step 2.		
Notes				

TP Id		TP/WAN/SEN/WSI/BP/BV-005		
TP label		Port Types		
Coverage	**Spec**	[OASIS/WS-I BP]		
	Testable items	BP-R2301; M		
Applicability		C_SEN_000		
Initial condition		The simulated receiver has a WebService enabled with many different services and the sender under test has a SOAP message ready to be sent to the respective service according to its needs.		
Test procedure		1. Wait until the sender under test sends a SOAP message or, if necessary, force it to send a SOAP message. 2. Take the WSDL description of the web service using its URL and check the wsdl:parts elements in the wsdl:message. 3. Compare their order with the soap:Body elements order.		
Pass/Fail criteria		In step 3, 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/SEN/WSI/BP/BV-006		
TP label		SOAP Binding		
Coverage	**Spec**	[OASIS/WS-I BP]		
	Testable items	BP-R2742; O	BP-R2743; O	
Applicability		C_SEN_000 AND C_SEN_WSI_034		
Initial condition		The simulated receiver has a WebService enabled with many different services and the sender under test has a SOAP message ready to be sent to the respective service according to its needs.		
Test procedure		1. Wait until the sender under test sends a SOAP message or, if necessary, force it to send a SOAP message. 2. The simulated receiver responds with a message that will cause that sender to generate a fault. 3. The sender under test sends a fault message. 4. Check the envelope's fault detail element and the SOAP header block's header processing fault.		
Pass/Fail criteria		In step 2, verify that the detail element cannot be described by the soapbind:fault element of the WSDL description, and that the header block cannot be described by a soapbind:headerfault element of the WSDL description.		
Notes				

TP Id		TP/WAN/SEN/WSI/BP/BV-006_B		
TP label		SOAP Binding 2		
Coverage	**Spec**	[OASIS/WS-I BP]		
	Testable items	BP-R2712; M	BP-R2735; M	BP-R2755; M
		BP-R2737; M	BP-R2738; M	BP-R2739; O
		BP-R2752; O	BP-R2753; O	
Applicability		C_SEN_000		
Initial condition		The simulated receiver has a WebService enabled with many different services and the sender under test has a SOAP message ready to be sent to the respective service according to its needs.		
Test procedure		1. Wait until the sender under test sends any SOAP message or, if necessary, force it to send any SOAP message. 2. Check the captured message.		
Pass/Fail criteria		Look into the WSDL description of the web service and check: • in step 2: o if the SOAP header block is not described in the wsdl:binding, it may 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; o that all soapbind:headers specified in wsdl:input or wsdl:output of a wsdl:operation of a wsdl:binding are included in the envelope; o if an rpc-literal binding is used; that the part accessor of the envelope has a local name equal to the name of the attribute of the wsdl:part element; that it is not placed in a namespace, and that its descendents have a namespace qualified by the schema in which the part accessor types are defined; o if a doc-literal binding is used, that the child element of the soap:Body is an instance of the global element declaration referenced by the corresponding wsdl:message part.		
Notes				

A.3 Subgroup 1.1.2 – Basic security profile (BSP)

TP Id		TP/WAN/SEN/WSI/BSP/BV-000		
TP label		TLS Ciphersuites		
Coverage	**Spec**	[OASIS/WS-I BSP]		
	Testable items	BSP-322; R	BSP-323; R	
	Spec	[b-CDG 2012], WAN Interface		
	Testable items	SecGuidelines2; M		
Applicability		C_SEN_000		
Initial condition		The simulated receiver has a WebService enabled with many different services and the sender under test has a SOAP message ready to be sent to the respective service according to its needs.		
Test procedure		1. If an instance is FIPS compliant (C_SEN_WSI_002=true): a. Load the simulated receiver supporting TLS_RSA_FIPS_WITH_AES_128_CBC_SHA. b. Make the sender under test establish a TLS connection. c. Check in the TLS handshake that the sender under test SHOULD not support: ❑ any ciphersuites with an DH_anon in their symbolic name ❑ any ciphersuites with a MD5 in their symbolic name ❑ any of the following ciphersuites: • TLS_RSA_WITH_NULL_SHA • TLS_RSA_WITH_NULL_MD5 ❑ any ciphersuites that use 40 or 56 bit keys. d. Check that the sender under test supports TLS_RSA_FIPS_WITH_AES_128_CBC_SHA e. Close the connection. 2. If an instance is not FIPS compliant (C_SEN_WSI_002=false): a. Load the simulated receiver supporting TLS_RSA_WITH_AES_128_CBC_SHA. b. Make the sender under test establish a TLS connection. c. Check in the TLS handshake that the sender under test does not support (these are recommendations only): ❑ any ciphersuites with an DH_anon in their symbolic name ❑ any ciphersuites with a MD5 in their symbolic name ❑ any of the following ciphersuites: • TLS_RSA_WITH_NULL_SHA • TLS_RSA_WITH_NULL_MD5 ❑ any ciphersuites that use 40 or 56 bit keys. d. Check that the sender under test supports: TLS_RSA_WITH_AES_128_CBC_SHA.		
Pass/Fail criteria		• If C_SEN_WSI_002 is supported, the sender under test must support TLS_RSA_FIPS_WITH_AES_128_CBC_SHA. • If C_SEN_WSI_002 is not supported, the sender under test must support TLS_RSA_WITH_AES_128_CBC_SHA. • The ciphersuites supported must match with these PICS: C_SEN_WSI_027, C_SEN_WSI_028, C_SEN_WSI_029, C_SEN_WSI_030.		
Notes				

TP Id		TP/WAN/SEN/WSI/BSP/BV-001		
TP label		Security Policy		
Coverage	**Spec**	[OASIS/WS-I BSP]		
	Testable items	BSP-R3105; O		
Applicability		C_SEN_000 AND C_SEN_WSI_003		
Initial condition		The simulated receiver has a WebService enabled with many different services. The sender under test and the simulated receiver have never been partners in a message exchange.		
Test procedure		1. Make the sender under test send its supported configuration to the receiver, including supported encryption and/or signatures and security tokens. 2. The simulated receiver waits for a SOAP message from the sender. 3. The simulated receiver checks the received message, ensuring that the sender agrees or disagrees in an out of band fashion with the receiver.		
Pass/Fail criteria		Step 3 is achieved.		
Notes		This is WS-Trust negotiation.		

TP Id		TP/WAN/SEN/WSI/BSP/BV-003		
TP label		Basic Profile Clarification		
Coverage	**Spec**	[OASIS/WS-I BSP]		
	Testable items	BSP-R5801; M	BSP-R5805; M	BSP-R5813; M
Applicability		C_SEN_000 AND C_SEN_WSI_003		
Initial condition		The simulated receiver has a WebService enabled with many different services and the sender under test has a SOAP message ready to be sent to the respective service according to its needs.		
Test procedure		1. Make the sender under test send a SOAP message using security. 2. As the simulated receiver knows its description (wsdl), after reversing the SOAP message security, check that: a. The order of the elements in the soap:body is the same as the wsdl:parts in the wsdl:message. b. The envelope includes all soapbind:headers specified on a wsdl:input or wsdl:output of a wsdl:operation of a wsdl:binding. c. If doc-literal binding is used, it 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.		
Pass/Fail criteria		All steps are as specified within the test procedure above.		
Notes		"Reversing SOAP Message Security" means removing the various impacts of applying "SOAP Message Security" that may have been applied since the MESSAGE (BP1.0) or ENVELOPE (BP 1.1) was originally created for that recipient according to the BP. This may mean decrypting relevant portions of the XML or removing XML signature elements or making other reverse transformations as appropriate to the aspects of SOAP message security that were applied in the specific circumstance.		

TP Id		TP/WAN/SEN/WSI/BSP/BV-005		
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		
Applicability		C_SEN_000 AND C_SEN_WSI_004		
Initial condition		The simulated receiver has a WebService enabled with many different services and the sender under test has a SOAP message ready to be sent to the respective service according to its needs.		
Test procedure		1. Make the sender under test send a SOAP message using a Timestamp element. 2. Check in the captured message that: a. 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 the Expires element is present, only one, it comes after the Created element and: ❑ 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.		
Pass/Fail criteria		The elements in step 2 are as specified within the test procedure above.		
Notes				

TP Id		TP/WAN/SEN/WSI/BSP/BV-006		
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	
Applicability		C_SEN_000 AND C_SEN_WSI_016		
Initial condition		The simulated receiver has a WebService enabled with many different services and the sender under test has a SOAP message ready to be sent to the respective service according to its needs.		
Test procedure		1. Make the sender under test send a SOAP message using a security token reference (STR) with an STR_Reference. <wsse:SecurityTokenReference wsu:Id="..."> <wsse:Reference URI="..." ValueType="..."/> </wsse:SecurityTokenReference> 2. Check in the captured message that: a. There is only one STR_Reference within the SECURITY_TOKEN_REFERENCE. b. STR_Reference does not reference another SECURITY_TOKEN_REFERENCE or an STR_Embedded. c. URI Attribute is present. d. ValueType attribute is present and it contains a value for the referenced security token specified by the corresponding security token profile (e.g., X.509 certificate token). e. SECURITY_TOKEN_REFERENCE does not contain an STR_KEY_NAME and does not reference a ds:KeyInfo element.		
Pass/Fail criteria		Check that SECURITY_TOKEN_REFERENCE is as specified in steps 1 and 2.		
Notes				

<table>
<tr><td colspan="2">TP Id</td><td colspan="3">TP/WAN/SEN/WSI/BSP/BV-007</td></tr>
<tr><td colspan="2">TP label</td><td colspan="3">Security Token References - Key Identifier</td></tr>
<tr><td rowspan="3">Coverage</td><td>Spec</td><td colspan="3">[OASIS/WS-I BSP]</td></tr>
<tr><td rowspan="2">Testable items</td><td>BSP-R3054; M</td><td>BSP-R3063; M</td><td>BSP-R3070; M</td></tr>
<tr><td>BSP-R3071; M</td><td></td><td></td></tr>
<tr><td colspan="2">Applicability</td><td colspan="3">C_SEN_000 AND C_SEN_WSI_017</td></tr>
<tr><td colspan="2">Initial condition</td><td colspan="3">The simulated receiver has a WebService enabled with many different services and the sender under test has a SOAP message ready to be sent to the respective service according to its needs.</td></tr>
<tr><td colspan="2">Test procedure</td><td colspan="3">1. Make the sender under test send a SOAP message using a security token reference (STR) with a key identifier reference:
<wsse:SecurityTokenReference>
<wsse:KeyIdentifier wsu:Id="..."
ValueType="..."
EncodingType="...">...
</wsse:KeyIdentifier>
</wsse:SecurityTokenReference>
2. 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 an SAML token is referenced, the encodingType attribute is not present.
c. If the referenced token is different from the SAML token, the encodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary".</td></tr>
<tr><td colspan="2">Pass/Fail criteria</td><td colspan="3">In step 2, attributes are as specified.</td></tr>
<tr><td colspan="2">Notes</td><td colspan="3"></td></tr>
</table>

TP Id		TP/WAN/SEN/WSI/BSP/BV-008		
TP label		Security Token References - Embedded References		
Coverage	**Spec**	[OASIS/WS-I BSP]		
	Testable items	BSP-R3060; M	BSP-R3025; M	BSP-R3056; M
Applicability		C_SEN_000 AND C_SEN_WSI_018		
Initial condition		The simulated receiver has a WebService enabled with many different services and the sender under test has a SOAP message ready to be sent to the respective service according to its needs.		
Test procedure		1. Make the sender under test send a SOAP message using a security token reference (STR) with an embedded reference: <wsse:SecurityTokenReference> <wsse:Embedded wsu:Id="..."> ... </wsse:Embedded> </wsse:SecurityTokenReference>> 2. Check in the captured message that: a. STR_Embedded has only one child element that is an internal security token, and it is in the same format as if it were a child of a security header. b. STR_Embedded does not contain a wsse:SecurityTokenReference child element.		
Pass/Fail criteria		In step 2, "Security Token Reference Embedded" are as specified.		
Notes		An internal token reference is a reference to a token that is contained in the same message. An example of an incorrect and a correct format are: INCORRECT: <!-- This example is incorrect because the wsse:Embedded element carries the data for the X.509 certificate directly rather than as a wsse:BinarySecurityToken element --> <wsse:SecurityTokenReference> <wsse:Embedded wsu:Id="SomeCert"> lui+Jy4WYKGJW5xM3aHnLxOpGVIpzSg4V486hHFe7sHET/uxxVBovT7JV1A2RnWSWkXm9jAEdsm/... </wsse:Embedded> </wsse:SecurityTokenReference> CORRECT: <wsse:SecurityTokenReference> <wsse:Embedded wsu:Id="TheEmbeddedElementAroundSomeCert"> <wsse:BinarySecurityToken wsu:Id='SomeCert' ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"> lui+Jy4WYKGJW5xM3aHnLxOpGVIpzSg4V486hHFe7sHET/uxxVBovT7JV1A2RnWSWkXm9jAEdsm/... </wsse:BinarySecurityToken> </wsse:Embedded> </wsse:SecurityTokenReference>		

TP Id		TP/WAN/SEN/WSI/BSP/BV-009		
TP label		Security Token References - Internal References		
Coverage	**Spec**	[OASIS/WS-I BSP]		
	Testable items	BSP-R3022; M	BSP-R3023; M	BSP-R5204; M
		BSP-R5205; M	BSP-R3067; M	
Applicability		C_SEN_000 AND C_SEN_WSI_019		
Initial condition		The simulated receiver has a WebService enabled with many different services and the sender under test has a SOAP message ready to be sent to the respective service according to its needs.		
Test procedure		1. Make the sender under test send a SOAP message including a SecurityTokenReference with an internal reference. 2. Check in the captured message that: a. The SECURITY_TOKEN_REFERENCE references an internal security token. b. The SECURITY_TOKEN_REFERENCE contains an STR_Reference or STR_Embedded. It is recommended to be an STR_Reference. c. The STR_Reference to an INTERNAL_SECURITY_TOKEN which has an ID attribute contains a URI attribute with a shorthand XPointer value. d. The INTERNAL_SECURITY_TOKEN precedes all SECURITY_TOKEN_REFERENCE elements that reference it in the SOAP envelope.		
Pass/Fail criteria		References are as specified within the test procedure above.		
Notes		The internal token reference is a reference to a token that is contained in the same message.		

TP Id		TP/WAN/SEN/WSI/BSP/BV-010		
TP label		Security Token References - External References		
Coverage	**Spec**	[OASIS/WS-I BSP]		
	Testable items	BSP-R3024; M		
Applicability		C_SEN_000 AND C_SEN_WSI_020		
Initial condition		The simulated receiver has a WebService enabled with many different services and the sender under test has a SOAP message ready to be sent to the respective service according to its needs.		
Test procedure		1. Make the sender under test send a SOAP message including a SecurityTokenReference with an external reference. 2. Check in the text file that: a. It is recommended that the external token reference contains an STR_Reference.		
Pass/Fail criteria		References are as specified within the test procedure above.		
Notes		The external token reference is a reference to a token that is not contained in the same message.		

<table>
<tr><td colspan="2">TP Id</td><td colspan="3">TP/WAN/SEN/WSI/BSP/BV-023</td></tr>
<tr><td colspan="2">TP label</td><td colspan="3">SAML Token</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-R6601; M</td><td>BSP-R6602; M</td><td>BSP-R6609; M</td></tr>
<tr><td>BSP-R6603; M</td><td>BSP-R6604; M</td><td>BSP-R6605; M</td></tr>
<tr><td>BSP-R6606: M</td><td>BSP-R6607; M</td><td>BSP-R6608; M</td></tr>
<tr><td colspan="2">Applicability</td><td colspan="3">C_SEN_000</td></tr>
<tr><td colspan="2">Initial condition</td><td colspan="3">The simulated receiver has a WebService enabled with many different services and the sender under test has a SOAP message ready to be sent to the respective service according to its needs.</td></tr>
<tr><td colspan="2">Test procedure</td><td colspan="3">1. Make the sender under test send a SOAP message using an SAML token.
2. Check in the captured message that the expected saml:Assertion element confirms that:
a. SAML KeyInfo does not contain a reference to an SAML token.
b. In an STR KeyIdentifier that references an SAML token:
❑ EncodingType attribute is not present.
❑ ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID".
❑ The Value encoded is an xs:string.
c. If a security token reference references an external SAML token:
❑ saml:AuthorityBinding element is present
❑ AuthorityKind = Value of saml:AssertionIdReference.</td></tr>
<tr><td colspan="2">Pass/Fail criteria</td><td colspan="3">The SAML token element is as specified within the test procedure above.</td></tr>
<tr><td colspan="2">Notes</td><td colspan="3"></td></tr>
</table>

A.4 Subgroup 1.1.3 – Reliable messaging (RM)

TP Id		TP/WAN/SEN/WSI/RM/BV-000		
TP label		Protocol Preconditions		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	Namespace; M	ProtocolPrec 2; M	
Applicability		C_SEN_000 AND C_SEN_WSI_021		
Initial condition		The sender under test and the simulated receiver are in the "None" sequence state.		
Test procedure		1. The sender under test sends a CreateSequence message with an offer element to the receiver. 2. The simulated receiver responds with a CreateSequenceResponse message accepting the offer. 3. The sender sends a Sequence message. 4. The receiver responds with its Sequence message and a SequenceAcknowledgement element. 5. The 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", and in step 1 the CreateSequence request is made.		
Notes				

TP Id		TP/WAN/SEN/WSI/RM/BV-001		
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	
Applicability		C_SEN_000 AND C_SEN_WSI_021 AND (C_SEN_WSI_023 OR C_SEN_WSI_024)		
Initial condition		The sender under test and the simulated receiver are in the "None" sequence state. The simulated receiver is able to avoid the response to a CreateSequence message.		
Test procedure		1. Make the sender send a CreateSequence message. 2. The simulated receiver does not respond to that message. 3. If C_SEN_WSI_023, the sender may retry transmission. 4. If C_SEN_WSI_024, the sender should retry transmission.		
Pass/Fail criteria		All steps are as specified within the test procedure above.		
Notes				

TP Id		TP/WAN/SEN/WSI/RM/BV-003		
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
Applicability		C_SEN_000 AND C_SEN_WSI_021		
Initial condition		The sender under test and the simulated receiver are in the "None" sequence state.		
Test procedure		1. The sender under test sends a CreateSequence message with an offer element. 2. The simulated receiver responds with CreateSequenceResponse accepting the offer. 3. The sender sends a Sequence message. 4. The receiver responds with a SOAP message including a SequenceAcknowledgement header block and a Sequence header block (indicating that it is the last message). 5. The sender responds including a SequenceAcknowledgement header block. ❑ If the SOAP message also contains a CloseSequence header block or any other header block (piggybacking), all the header blocks will have the same EPR (endpoint reference). ❑ If not, any other header block is sent in the same SOAP message, the sender under test sends a message for every other RM-element (not piggybacking).		
Pass/Fail criteria		In step 5, If the sender sends only one message with more than one header block (piggybacking), the EPR is the same for every header block.		
Notes		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 by there only being one "wsa:To" element in the soap:header.		

TP Id		TP/WAN/SEN/WSI/RM/BV-004		
TP label		Sequence Creation		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	WSAddress 1; C	SeqCreation 1; M	SeqCreation 2; O
		SeqCreation 5; M	SeqCreation 7; M	SeqCreation 8; M
		SeqCreation 9; O	SeqCreation 10; M	SeqCreation 11; M
		SeqCreation 12; M	SeqCreation 14; O	SeqCreation 15; O
		SeqCreation 22; O		
Applicability		C_SEN_000 AND C_SEN_WSI_021		
Initial condition		The sender under test and the simulated receiver are in the "None" sequence state.		
Test procedure		1. Wail until the sender under test sends a CreateSequence message. 2. Check that the captured message has the following properties: a. In the header block: ❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/CreateSequence. ❑ wsrm:CreateSequence is not present. b. In the body of the message: ❑ wsrm:AcksTo of type wsa:EndpointReferenceType is present and defines a valid endpoint. ❑ wsrm:Expires element, if present: • its type is xs:duration. ❑ If an offer element is present: • wsrm:IncompleteSequenceBehaviour element may be present. Possible values are: "discard", "DiscardEntireSequence", "DiscardFollowingFirstGap" and "NoDiscard". • wsrm:Identifier value is an absolute URI that uniquely identifies the offered Sequence. • wsrm:Expires element, if present, its type is xs:duration. • wsrm:Endpoint element is present and its type is wsa:EndpointReferenceType, and it defines a valid endpoint. 3. The simulated receiver responds using a CreateSequenceResponse message without an accept element or a CreateSequenceRefused fault. 4. If an offer element is present: ❑ The sender can reclaim the resources.		
Pass/Fail criteria		All elements are as specified within the test procedure above.		
Notes				

<table>
<tr><td colspan="2">TP Id</td><td colspan="3">TP/WAN/SEN/WSI/RM/BV-005</td></tr>
<tr><td colspan="2">TP label</td><td colspan="3">Closing a Sequence</td></tr>
<tr><td rowspan="4">Coverage</td><td>Spec</td><td colspan="3">[OASIS WS-I RM]</td></tr>
<tr><td rowspan="3">Testable items</td><td>WSAddress 1; C</td><td>SeqClosing 1; O</td><td>SeqClosing 2; M</td></tr>
<tr><td>SeqClosing 4; R</td><td>SeqClosing 8; O</td><td>SeqClosing 9; M</td></tr>
<tr><td>SeqClosing 10; R</td><td></td><td></td></tr>
<tr><td colspan="2">Applicability</td><td colspan="3">C_SEN_000 AND C_SEN_WSI_021 AND C_SEN_WSI_032</td></tr>
<tr><td colspan="2">Initial condition</td><td colspan="3">The sender under test and the simulated receiver are in the "Created" sequence state.</td></tr>
<tr><td colspan="2">Test procedure</td><td colspan="3">1. The sender under test starts to send a Sequence message including an AckRequested element or indicating that it is the last message in the header block of the last message.
2. The simulated receiver accepts all messages and if an offer was sent by the sender, it also sends a Sequence message indicating that it is the last message.
3. The sender sends with a SequenceAcknoledgement message.
4. If the sender sends a CloseSequenceMessage then check the received message:
a. In the header block:
❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/CloseSequence.
b. In the body of the message, within the CloseSequence element:
❑ wsrm:Identifier value = an absolute URI of the closing sequence.
❑ The presence of wsrm:LastMsgNumber is recommended, and if it is present it must be the same in all CloseSequence elements of that closing sequence.
5. Or else, if C_SEN_WSI_032 = TRUE then force the sender to close the sequence and check the received message.
a. In the header block:
❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/CloseSequence.
b. In the body of the message, within the CloseSequence element:
❑ wsrm:Identifier value = an absolute URI of the closing sequence.
❑ The presence of wsrm:LastMsgNumber is recommended, and if it is present it must be the same in all CloseSequence elements of that closing sequence.
6. The simulated receiver responds with a CloseSequenceResponse.</td></tr>
<tr><td colspan="2">Pass/Fail criteria</td><td colspan="3">All elements are as specified within the test procedure above.</td></tr>
<tr><td colspan="2">Notes</td><td colspan="3"></td></tr>
</table>

TP Id		TP/WAN/SEN/WSI/RM/BV-005_B		
TP label		Closing a Sequence Response		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	WSAddress 1; C	SeqClosing 1; O	SeqClosing 11; M
		SeqClosing 12; M		
Applicability		C_SEN_000 AND C_SEN_WSI_021 AND NOT(C_SEN_WSI_032)		
Initial condition		The sender under test and the simulated receiver are in the "Created" sequence state.		
Test procedure		1. Run the sender under test (make sure that the sender has something, a measure or anything else, to send). 2. Wait until the sender sends a CreateSequence message. 3. The simulated receiver responds with a CreateSequenceResponse. If an offer is sent by the sender in step 2, the receiver accepts the offer. 4. The sender under test starts to send a Sequence message including an AckRequested element or indicating that it is the last message in the header block of the last message. 5. The simulated receiver accepts all messages and if an offer was sent by the sender, it also sends a Sequence message indicating that it is the last message. 6. The sender sends with a SequenceAcknoledgement message. 7. The simulated receiver sends a CloseSequence element in the body of the message, including a correct LastMessageNumber. 8. The sender responds with a CloseSequenceResponse message including: a. In the header block: ❑ 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 response.		
Pass/Fail criteria		All elements are as specified within the test procedure above.		
Notes				

TP Id		TP/WAN/SEN/WSI/RM/BV-006		
TP label		Sequence Termination		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	WSAddress 1; M	SeqTermination 1; R	SeqTermination 2; M
		SeqTermination 4; O	SeqTermination 5; M	SeqTermination 7; M
		SeqTermination 11; M	SeqTermination 12; R	
Applicability		C_SEN_000 AND C_SEN_WSI_021 AND C_SEN_WSI_033		
Initial condition		The sender under test and the simulated receiver are in the "Created" sequence state.		
Test procedure		1. The sender sends Sequence messages including an AckRequested element or indicating that it is the last message in the header block of the last message. 2. The receiver under test responds using a SequenceAcknowledgement header block, accepting all messages. 3. If the sender under test sends a TerminateSequence element in the body of the message, the expected messages are: a. In the header block: ❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/TerminateSequence ❑ wsrm: TerminateSequence is not present. b. In the body of the message, within the TerminateSequence element: ❑ wsrm:Identifier value is an absolute URI of the terminating sequence. ❑ It is recommended that a LastMsgNumber element is present, and, if present, it must be equal to the LastMsgNumber of any CloseSequence message. 4. If the sender has sent a TerminateSequence element, the simulated receiver responds with a TerminateSequenceResponse message, including its Identifier element as an absolute URI. 5. Once the sequence is terminated, the sender under test does not send any message referencing that terminated sequence.		
Pass/Fail criteria		All elements are as specified within the test procedure above.		
Notes				

TP Id		TP/WAN/SEN/WSI/RM/BV-006_B		
TP label		Sequence Termination Response		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	WSAddress 1; M	SeqTermination 10; M	SeqTermination 13; M
		SeqTermination 14; M	SeqTermination 15; M	
Applicability		C_SEN_000 AND C_SEN_WSI_021 AND NOT(C_SEN_WSI_033)		
Initial condition		The sender under test and the simulated receiver are in the "Created" sequence state.		
Test procedure		1. The sender sends Sequence messages including an AckRequested element or indicating that it is the last message in the header block of the last message. 2. The receiver under test responds using a SequenceAcknowledgement header block, accepting all messages. 3. The simulated receiver sends a TerminateSequence element in the body of the message, with a correct LastMsgNumber. 4. The sender responds only with a message including: a. In the header block: ❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/TerminateSequenceResponse ❑ wsrm: TerminateSequenceResponse is not present. b. In the body of the message within the TerminateSequenceResponse element: ❑ wsrm:Identifier element as an absolute URI of the terminating sequence. 5. Once the sequence is terminated, the sender under test does not send any message referencing that terminated sequence.		
Pass/Fail criteria		All elements are as specified within the test procedure above.		
Notes				

TP Id		TP/WAN/SEN/WSI/RM/BV-007		
TP label		Sequences		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	ProtocolInv 1; M	Sequences 1; M	Sequences 2; M
		Sequences 3; M	Sequences 5; M	Sequences 6; M
		Sequences 7; M	Sequences 8; M	
Applicability		C_SEN_000 AND C_SEN_WSI_021		
Initial condition		The sender under test and the simulated receiver are in the "Created" sequence state.		
Test procedure		1. Wait until the sender under test sends Sequence message/s including an AckRequested element or indicating that it is the last message in the last message header block. 2. The expected message/s are: ❑ wsrm:MessageNumber element is of type MessageNumberType and starts in 1 and increments by 1 in every sequential message. ❑ There is only one Sequence header block in each message. ❑ wsrm:Identifier element must be present in the header block and must be an absolute URI that uniquely identifies the sequence. ❑ mustUnderstand attribute = "1" or "true". 3. The simulated receiver responds using a SequenceAcknowledgement header block accepting all messages received.		
Pass/Fail criteria		All elements are as specified in step 2.		
Notes				

TP Id		TP/WAN/SEN/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
Applicability		C_SEN_000 AND C_SEN_WSI_021 AND C_SEN_WSI_034		
Initial condition		The sender under test and the simulated receiver are in the "None" sequence state. The simulated receiver is able to send a CloseSequence message in the "None" sequence state.		
Test procedure		1. The simulated receiver transmits a CloseSequence message with an unknown identifier. 2. The sender under test generates an UnknownSequence fault. It is recommended that the fault is transmitted to the receiver. 3. That message includes the following properties: ❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/fault ❑ Code = Sender ❑ Subcode = wsrm:UnknownSequence ❑ Reason = The value if wsrm:Identifier is not a known Sequence identifier ❑ Detail = <wsrm:Identifier...> xs:anyURI </wsrm:Identifier>.		
Pass/Fail criteria		All elements are as specified in step 3.		
Notes				

TP Id		TP/WAN/SEN/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	
Applicability		C_SEN_000 AND C_SEN_WSI_021 AND C_SEN_WSI_034		
Initial condition		The sender under test and the simulated receiver are in the "Created" sequence state.		
Test procedure		1. The sender under test starts to send Sequence messages with their respective message number. 2. Wait until the sender sends an AckRequested element or indicates that the message is the last one. 3. The simulated receiver responds with a SequenceAcknowledgement with an AckRange, a None and a Nack element. 4. The sender generates an InvalidAcknowledgement fault. It is recommended that the fault is transmitted to the receiver. 5. That message includes the following properties: ❑ wsa:Action = http://docs.oasis-open.org/ws-rx/wsrm/200702/fault ❑ Code = Sender ❑ Subcode = wsrm:InvalidAcknowledgement ❑ Reason = <any> ❑ Detail = <any related to the message that produces the fault>.		
Pass/Fail criteria		All elements are as specified in step 5.		
Notes				

TP Id		TP/WAN/SEN/WSI/RM/BV-012		
TP label		Message Number Rollover		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	MessageNumrRoll 4; R		
Applicability		C_SEN_000 AND C_SEN_WSI_021		
Initial condition		The sender under test and the simulated receiver are in the "Created" sequence state. The simulated receiver is able to send a MessageNumberRollover fault instead of a SequenceAcknowledgement message.		
Test procedure		1. The sender under test transmits a Sequence message. 2. The simulated receiver generates a MessageNumberRollover fault, which is transmitted to the sender. 3. The sender should retransmit undelivered messages until the receiver closes or terminates the sequence.		
Pass/Fail criteria		The sender should retransmit undelivered messages in step 3.		
Notes				

TP Id		TP/WAN/SEN/WSI/RM/BV-012_A		
TP label		Create Sequence Refused		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	SeqRefused 3; M		
Applicability		C_SEN_000 AND C_SEN_WSI_021		
Initial condition		The sender under test and the simulated receiver are in the "None" sequence state. The simulated receiver is able to send a CreateSequenceRefused fault instead of a CreateSequenceResponse message.		
Test procedure		1. Wait until the sender under test sends a CreateSequence message to the simulated receiver. 2. The simulated receiver responds with a CreateSequenceRefused fault. 3. The sender must terminate the sequence.		
Pass/Fail criteria		The sender terminates the sequence when it receives a CreateSequenceRefused fault.		
Notes				

TP Id		TP/WAN/SEN/WSI/RM/BV-012_B		
TP label		Sequence Closed Fault		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	SeqClosedFault 3; M		
Applicability		C_SEN_000 AND C_SEN_WSI_021		
Initial condition		The sender under test and the simulated receiver are in the CreatedSequence state. The simulated receiver is able to send a SequenceClosed fault instead of a SequenceAcknowledgement message.		
Test procedure		1. The sender under test sends a sequence to the simulated receiver sending an AckRequested message or indicating that it is the last message. 2. The simulated receiver sends a SequenceClosed fault. 3. The sender must close the sequence.		
Pass/Fail criteria		The sender closes the sequence when it receives a SequenceClosed fault.		
Notes				

TP Id		TP/WAN/SEN/WSI/RM/BV-015		
TP label		Securing Sequences Using WS-Security		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	SecSeqWSS 5; R	SecSeqWSS 6; R	
Applicability		C_SEN_000 AND C_SEN_WSI_021 AND C_SEN_WSI_003		
Initial condition		The sender under test and the simulated receiver are in the "None" sequence state.		
Test procedure		1. Wait until the sender under test sends a CreateSequence message. 2. It is recommended that the received message includes a UsesSequenceSTR element in the header block. If the element is included, it MUST include a soap:mustUnderstand attribute = "true".		
Pass/Fail criteria		The recommended element in step 2 is as specified within the test procedure above.		
Notes				

TP Id		TP/WAN/SEN/WSI/RM/BV-016		
TP label		Securing Sequences Using SSL/TLS		
Coverage	**Spec**	[OASIS WS-I RM]		
	Testable items	SecSeqSSL/TLS 1; M	SecSeqSSL/TLS 2; O	SecSeqSSL/TLS 3; M
Applicability		C_SEN_000 AND C_SEN_WSI_021		
Initial condition		The sender under test and the simulated receiver are in the "None" sequence state.		
Test procedure		1. Wait until the sender under test sends a CreateSequence message. 2. If the sender binds a sequence to the underlying SSL/TLS sessions(s) it includes the UsesSequenceSSL element as a SOAP header block within the CreateSequence message, with a soap:mustUnderstand attribute = "true".		
Pass/Fail criteria		If the sender binds the sequence to the underlying TSL session, elements are as specified in step 2.		
Notes				

Bibliography

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

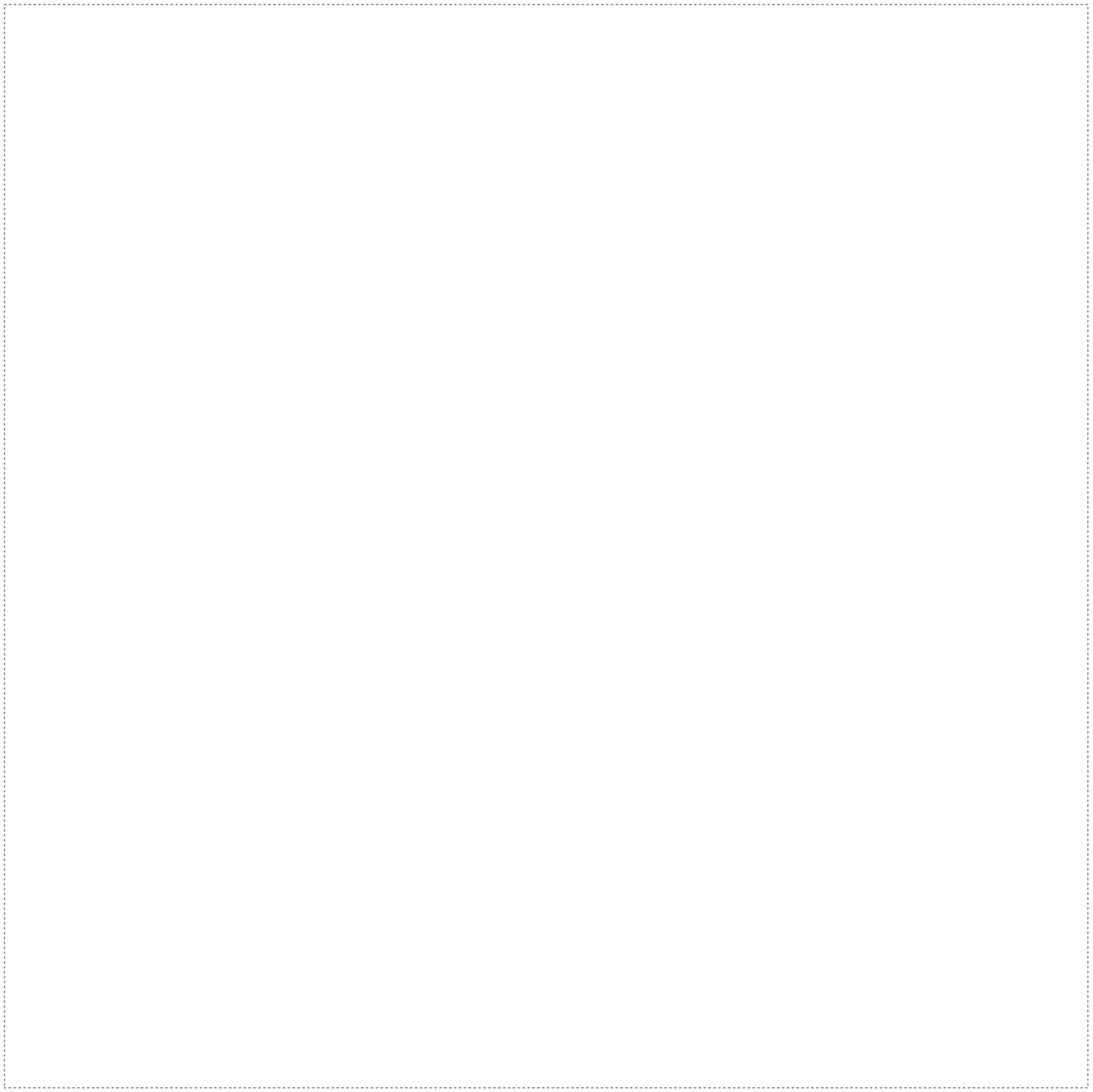
[b-CDG 2010] Continua Health Alliance (2010), *Continua Design Guidelines v1.5*.

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

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

[b-ETSI SR 001 262] ETSI SR 001 262 v1.8.1 (2003), *ETSI drafting rules*.

[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	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
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2015