

BS ISO/IEC 18450:2013



BSI Standards Publication

Information technology — Telecommunications and information exchange between systems — Web Services Description Language (WSDL) for CSTA Phase III

bsi.

...making excellence a habit.
AVA Documents™
Document Sharing Hub

National foreword

This British Standard is the UK implementation of ISO/IEC 18450:2013.

The UK participation in its preparation was entrusted to Technical Committee ICT/-/1, Information systems co-ordination.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2013.
Published by BSI Standards Limited 2013

ISBN 978 0 580 79977 8

ICS 35.060

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 March 2013.

Amendments issued since publication

Date	Text affected
------	---------------

INTERNATIONAL STANDARD

ISO/IEC
18450

First edition
2013-03-01

Information technology — Telecommunications and information exchange between systems — Web Services Description Language (WSDL) for CSTA Phase III

*Technologies de l'information — Télécommunications et échange
d'information entre systèmes — Langage de description des services
Web (WSDL) pour CSTA Phase III*



Reference number
ISO/IEC 18450:2013(E)

AVA **Documents**
Document Sharing Hub
© ISO/IEC 2013



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2013

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword	vi
Introduction.....	vii
WSDL Tools Acknowledgement	vii
1 Scope	1
2 Conformance	1
3 Normative references	2
3.1 Ecma references	2
3.2 World Wide Web Consortium references.....	2
4 Terms and definitions	2
5 Web Services Overview	2
6 WSDL Overview	4
6.1 Namespaces.....	4
6.2 Data Types	4
6.3 Messages	5
6.3.1 Message Names.....	5
6.3.2 Message Parts	5
6.3.3 Negative Response Message	5
6.3.4 Example.....	5
6.4 Operations.....	6
6.4.1 Switching Function WSDL.....	6
6.4.2 Computing Function WSDL.....	7
6.4.3 Summary of CSTA Request/Response & Event Models and WSDL Operation Types.....	8
6.5 Port Types	8
6.5.1 Example.....	8
6.6 Bindings	10
6.7 Port.....	10
6.8 Service	10
7 Capability Exchange Services – Messages and Operations.....	10
7.1 Messages	10
7.2 Operations.....	13
8 System Services – Messages and Operations	14
8.1 Messages	14
8.2 Operations.....	18
9 Monitoring Services – Messages and Operations	20
9.1 Messages	20
9.2 Operations.....	22
10 Snapshot Services – Messages and Operations	23
10.1 Messages	23
10.2 Operations.....	25
11 Call Control Features – Messages and Operations	26
11.1 Messages	26
11.2 Operations.....	41
12 Call Associated Features - Messages and Operations.....	49
12.1 Messages	49
12.2 Operations.....	53

13	Media Attachment Features - Messages and Operations.....	55
13.1	Messages.....	55
13.2	Operations.....	56
14	Routeing Features - Messages and Operations	57
14.1	Messages.....	57
14.2	Operations.....	60
15	Physical Device Features - Messages and Operations	61
15.1	Messages.....	61
15.2	Operations.....	73
16	Logical Device Features - Messages and Operations.....	79
16.1	Messages.....	79
16.2	Operations.....	90
17	Device Maintenance Events - Messages and Operations	97
17.1	Messages.....	97
17.2	Operations.....	98
18	I/O Services Features - Messages and Operations	98
18.1	Messages.....	98
18.2	Operations.....	103
19	Data Collection Features - Messages and Operations.....	107
19.1	Messages.....	107
19.2	Operations.....	110
20	Voice Features - Messages and Operations	112
20.1	Messages.....	112
20.2	Operations.....	123
21	Call Detail Record Features - Messages and Operations.....	129
21.1	Messages.....	129
21.2	Operations.....	131
22	Location Services - Messages and Operations	132
22.1	Messages.....	132
22.2	Operations.....	137
23	Vendor Specific Features - Messages and Operations.....	140
23.1	Messages.....	140
23.2	Operations.....	142
24	Basic Telephony Profile – Port Type Pair	143
25	Routeing Profile - Port Type Pair	146
26	Level 1a Voice Browser Profile - Port Type Pair.....	147
27	Level 1b Voice Browser Profile - Port Type Pair	149
28	Level 2a Voice Browser Profile - Port Type Pair.....	151
29	Level 2b Voice Browser Profile - Port Type Pair	153
30	Minimal uaCSTA Call Control Profile – Port Type	156
31	Basic uaCSTA Call Control Profile – Port Type Pair	158
32	Advanced uaCSTA Call Control Profile – Port Type Pair	161
33	Conferencing uaCSTA with Basic Call Control Profile – Port Type Pair	166
34	Conferencing uaCSTA with Advanced Call Control Profile – Port Type Pair.....	170
35	Basic uaCSTA Device Feature with Basic Call Control Profile – Port Type Pair	176
36	Basic uaCSTA Device Feature with Advanced Call Control Profile – Port Type Pair.....	180
37	Speaker uaCSTA Device Feature with Basic Call Control Profile – Port Type Pair.....	186

38	Speaker uaCSTA Device Feature with Advanced Call Control Profile – Port Type Pair	190
Annex A	(normative) Event Channel Establishment	196
A.1	General	196
A.2	Event Channel from Switching Function	197
A.2.1	ISO/IEC 18051 7.1 Implicit Association	198
A.2.2	ISO/IEC 18051 7.2 Implicit Association	198
A.3	Event Channel from Session.....	198
A.4	Event Channel from Event Sources in Switching Function (Option).....	198
Annex B	(informative) Example SOAP 1.1/HTTP Binding	201
Annex C	(informative) WSDL Considerations	203

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 18450 was prepared by Ecma International (as ECMA-348) and was adopted, under a special “fast-track procedure”, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

Introduction

This International Standard defines a Web Services Description Language (WSDL) for Phase III of Computer Supported Telecommunications Applications (CSTA).

This International Standard is part of a Suite of Standards and Technical Reports for Phase III of CSTA. All of the Standards and Technical Reports in the Suite are based upon the practical experience of Ecma member companies and each one represents a pragmatic and widely based consensus.

This first Edition of ISO/IEC 18450 (ECMA-348) is fully aligned with ISO/IEC 18051 6th Edition and it imports the CSTA XML schemata from ISO/IEC 18056 5th Edition.

This edition introduces WS-Eventing to establish event channels and implicit associations, and recasts outbound operations into inbound operations in the Computing Function WSDL. This change is fully conformant to WS-I and supports SOAP/HTTP as a complete binding of the WSDL.

WSDL Tools Acknowledgement

The WSDL definition files, for the computing function, switching function and common messages, reflecting the WSDL specifications in this International Standard have been verified using the XML SPY tool.

Information technology — Telecommunications and information exchange between systems — Web Services Description Language (WSDL) for CSTA Phase III

1 Scope

Web Services Description Language (WSDL) is an XML schema for describing Web services and how they can be accessed by Web based applications. WSDL facilitates the creation and deployment of web based applications. For example, by using WSDL with many industry Web services development environments, a web services developer can access features provided by an implementation without knowing details of the network or underlying transport protocols.

This International Standard specifies two WSDL documents: Computing Function WSDL and Switching Function WSDL, for ISO/IEC 18051, Services for Computer Supported Telecommunications Applications (CSTA) Phase III. All CSTA features (e.g., services and events) specified in ISO/IEC 18051 are specified in this International Standard. This Standard specifies pairs of port types, one from the Computing Function WSDL and one from the Switching Function WSDL, for all the profiles specified in ISO/IEC 18051.

A full WSDL document contains both abstract definitions (WSDL messages and port type elements) and concrete protocol specific definitions (WSDL bindings, ports, and service elements). This CSTA WSDL Standard specifies only the abstract definitions of a WSDL document. The concrete and protocol specific definitions are implementation specific and are outside the scope of this International Standard. Annex B illustrates an example SOAP over HTTP binding.

WSDL is defined in XML and XML Schemata. This International Standard builds upon the XML data types and imports all message formats specified in ISO/IEC 18056.

Annex A specifies mechanisms to establish event channels based on WS-Eventing. The event channels allow the Switching Function and Computing Function to be both a Service Requester and a Service Provider.

2 Conformance

In addition to the conformance requirements specified in Clause 2 of ISO/IEC 18051 and Clause 2 of ISO/IEC 18056, a conformant implementation supports at least one pair of the port types defined in this International Standard. Each port type pair in this International Standard corresponds to a profile specified in 2.3.1 of ISO/IEC 18051.

To support one pair of these port types, an implementation must support all the WSDL operations specified in the port type pair. A conformant implementation supports an operation by adhering to the corresponding service definition in ISO/IEC 18051 and by using the message formats and data types as specified in ISO/IEC 18056.

A conformant implementation also supports the event channel establishment specified in Annex A.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

3.1 Ecma references

ISO/IEC 18051:2012, *Information technology — Telecommunications and information exchange between systems — Services for Computer Supported Telecommunications Applications (CSTA) Phase III (ECMA-269)*

ISO/IEC 18056:2012, *Information technology — Telecommunications and information exchange between systems — XML Schema Definitions for Computer Supported Telecommunications Applications (CSTA) Phase III (ECMA-323)*

ISO/IEC 25437:2012, *Information technology — Telecommunications and information exchange between systems — WS-Session — Web services for application session services (ECMA-366)*

ISO/IEC TR 18053:2000, *Information technology — Telecommunications and information exchange between systems — Glossary of definitions and terminology for Computer Supported Telecommunications Applications (CSTA) Phase III (ECMA TR/72)*

3.2 World Wide Web Consortium references

SOAP 1.1, Simple Object Access Protocol (SOAP) 1.1, W3C Note 08 May 2000

XML 1.0, Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation 06 October 2000

XML Schema 1.0, XML Schema Language Part 1: Structures. W3C Recommendation 2 May 2001 and XML Schema Language Part 2: Datatypes. W3C Recommendation 2 May 2001

WSDL 1.1, Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001

WS-Addressing, Web Services Addressing 1.0 – Core, ISO/IEC 40240
 Web Services Addressing 1.0 - SOAP Binding, ISO/IEC 40250
 Web Services Addressing 1.0 – Metadata, ISO/IEC 40260

WS-Eventing, Web Services Eventing (WS-Eventing) W3C Recommendation 13 December 2011

4 Terms and definitions

CSTA-specific terminology is defined in ISO/IEC TR 18053.

5 Web Services Overview

A Web service is the name for a service that is accessed by applications over the Internet. Conceptually this corresponds to a CSTA application (client) invoking services provided by a CSTA implementation (server) in a distributed environment.

Web services are more formally defined as: “encapsulated, loosely coupled contracted functions offered via standard protocols” where:

- Encapsulated means the implementation of the function is never seen from the outside
- Loosely coupled means changing the implementation of one function does not require a change of the invoking function and
- Contracted means there are publicly available descriptions of the function's behaviour, how to bind to the function as well as its input and output parameters.

The goal of Web services is a seamless integration of services across the different technologies (networks, software, hardware, etc.) available on the Internet. Web services require:

1. A standard way to represent data
2. A common, extensible message format
3. A common, extensible service description language
4. A common way to discover service providers

The first two requirements, a standard way to represent data and a common extensible message format, have already been achieved with ISO/IEC 18056. The ISO/IEC 18056 Standard includes extensible XML schemas for all of the categories of services defined in ISO/IEC 18051.

NOTE This International Standard satisfies the third requirement, a common, extensible service description language, by providing WSDL definitions for CSTA.

UDDI (Universal Discovery, Description, and Integration Registry) and other procedures to satisfy the fourth requirement are outside the scope of this International Standard.

There are three components involved with Web services:

- Service Requester – provides the following major functions: 1) optionally queries a Service Broker for where to find a suitable Service Provider and 2) binds and interacts with a Service Provider.
- Service Provider – provides the following major functions: 1) publishes its services (e.g. WSDL document) and 2) interacts with a Service Requester.
- Service Broker – acts as a look up service between a Service Provider and a Service Requester. A Service Broker is an optional component of a Web service.

For the purposes of this International Standard:

- A CSTA Service Requester is a Service Requester.
- A CSTA Service Provider is a Service Provider.
- A Service Broker is outside the scope of this International Standard.
- A WS-SF is a Switching Function that acts as a CSTA Service Requester and Provider.
- A WS-CF is a Computing Function that acts as a CSTA Service Requester and Provider.

6 WSDL Overview

WSDL enables applications to communicate with each other in an automated way. A WSDL document describes Web services both in an abstract (network independent) and concrete (with network protocol bindings, e.g. SOAP) way.

A WSDL document consists of sets of definitions. There are six major elements in a WSDL document:

- types – provides the data type definitions used to describe the messages exchanged
- message – defines the messages contents
- operation – defines how input and output messages are associated with a logical operation
- portType – groups a set of operations together
- binding – specifies concrete protocol and data format specifications for the operations and messages for a particular portType
- port – specifies an address for a binding
- service – used to group a set of related ports

For interoperability, this International Standard disallows WSDL 1.1 Notification and Solicit-response operations. Instead, this Standard uses a Switching Function WSDL to represent the Switching Function Services defined in ISO/IEC 18051 and a Computing Function WSDL to represent the CSTA Event Reports and Computing Function Services defined in ISO/IEC 18051. A Computing Function WSDL port type and a Switching Function WSDL port type that collectively specify a CSTA profile are paired by using this naming convention: the name of the Computing Function port type is the name of the paired Switching Function port type prefixed by string “CF-”.

6.1 Namespaces

There are two CSTA specific namespaces used in this International Standard:

- <http://www.ecma-international.org/standards/ecma-323/csta/ed6> - this namespace, for which “csta” is a shorthand, contains all of the CSTA XML messages specified in ISO/IEC 18056, 5th Edition. See 6.2.
- <http://www.ecma-international.org/standards/ecma-348/csta-wsdl/ed5> - this namespace, for which “tns” is a shorthand, contains the CSTA WSDL schema definitions specified in this International Standard.

NOTE The namespace for csta is appropriate for all of the XML Tag Options specified in ISO/IEC 18056.

There are also several other namespaces referenced by the CSTA WSDL schema such as:

- <http://schemas.xmlsoap.org/wsdl/> - this contains the W3C WSDL 1.1 schema
- <http://schemas.xmlsoap.org/wsdl/soap> - this contains the W3C SOAP bindings for WSDL 1.1
- <http://schemas.xmlsoap.org/wsdl/http> - this contains the W3C HTTP bindings for WSDL 1.1
- <http://www.w3.org/2001/XMLSchema> - this contains the W3C XML Schema Definition

6.2 Data Types

Data types describe the format of the XML elements in a message. Examples of data types are integers and strings.

This International Standard imports the CSTA XML data types standardized in ISO/IEC 18056.

6.3 Messages

Message definitions specify the contents of an XML message. The message definitions are used in a WSDL operation where they will be typed as an input or output message in the context of the WSDL operation.

This International Standard imports the CSTA XML elements specified in ISO/IEC 18056.

An example of an XML element specified in ISO/IEC 18056 is:

```
<xsd:element name="AnswerCall">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="callToBeAnswered" type="csta:ConnectionID"/>
      <xsd:element ref="csta:correlatorData" minOccurs="0"/>
      <xsd:element ref="csta:userData" minOccurs="0"/>
      <xsd:element ref="csta:extensions" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

The above definition, along with all ISO/IEC 18056 message definitions, is imported by the CSTA WSDL schema and is available in the <http://www.ecma-international.org/standards/ecma-323/csta/ed5> namespace.

6.3.1 Message Names

The WSDL message element specifies the name of a message that is referenced in a WSDL operation element. The naming convention used in this International Standard is to derive the WSDL message element name from the corresponding ISO/IEC 18056 element name. A lower case letter is used to differentiate the WSDL message name from the ISO/IEC 18056 element name.

For example, the ISO/IEC 18056 element name `AnswerCall` is used in the definition of the CSTA WSDL `answerCall` message name.

6.3.2 Message Parts

The WSDL message part element is used to define the actual format of the message. Although there can be multiple parts in a WSDL message, CSTA WSDL uses only one part and therefore can always use the same name (i.e. "parameter") for the message part name. The element type refers to an element defined in ISO/IEC 18056.

6.3.3 Negative Response Message

The negative response message is specified as follows:

```
<message name="negResponse">
  <part name="parameter" element="csta:CSTAErrorCode"/>
</message>
```

6.3.4 Example

The following example defines a WSDL message element definition for the ISO/IEC 18056 `AnswerCall` request message. Its name "answerCall" will be used to define the `AnswerCall` Operation.

```
<message name="answerCall">
  <part name="parameter" element="csta:AnswerCall"/>
</message>
```

6.4 Operations

A WSDL operation specifies the input message (if any), the output message (if any), and the fault message (if any) that is associated with the WSDL operation.

This formalizes the relationship between the CSTA XML elements standardized in ISO/IEC 18056. Although a person can determine the relationship between messages associated with an operation in ISO/IEC 18056 by its logical place in the ISO/IEC 18056 Standard and in some cases by naming conventions, CSTA WSDL specifies this relationship in a formal notation that can be understood by applications.

The operation name is derived from the feature name as specified in ISO/IEC 18056 and the message names are defined by the WSDL message element names.

The following subsections describe the types of WSDL operations used in the Switching Function WSDL and the Computing Function WSDL.

6.4.1 Switching Function WSDL

A request-response operation type is used when a Switching Function receives a message and responds with a corresponding message.

It is defined using one input message, one output message, and a fault message.

The CSTA Answer Call service uses this type of operation. The operation for CSTA Answer Call is shown below.

```
<operation name="CSTA-Answer-Call">  
  <input message="tns:answerCall"/>  
  <output message="tns:answerCallResponse"/>  
  <fault name="FaultName" message="tns:negResponse"/>  
</operation>
```

The following notation is used to describe a WSDL request-response operation type:

CSTA-Answer-Call	
input	tns:answerCall
output	tns:answerCallResponse
fault	tns:negResponse

A one-way operation type is used when a Switching Function receives a message and does not respond to the message.

This type of WSDL operation is defined using one input message.

The CSTA-Route-Select service uses this type of operation. The operation for CSTA-Route-Select is shown below.

```
<operation name="CSTA-Route-Select">  
  <input message="tns:routeSelect"/>  
</operation>
```

The following notation is used to describe a WSDL one-way operation type:

CSTA-Route-Select	
input	tns:routeSelect

WSDL 1.1 does not specify a fault message with a one-way operation type, however ISO/IEC 18051 specifies that a negative acknowledgement can be sent for certain services that do not specify a positive acknowledgement. Therefore a separate operation in the Computing Function WSDL is specified in this International Standard that provides a negative acknowledgement. It is not required to send a negative acknowledgement in these cases.

```
<operation name="CSTA-Fault-One-Way">
  <input message="tns:negResponse"/>
</operation>
```

6.4.2 Computing Function WSDL

A request-response operation type is used when a Switching Function sends a message and receives a corresponding response to the message.

The CSTA Call Detail Record Report service uses this type of operation. The operation is shown below.

```
<operation name="CSTA-CDR-Report">
  <input message="tns:cDRReport"/>
  <output message="tns:cDRReportResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
```

The following notation is used to describe a Computing Function WSDL request-response operation type:

CSTA-CDR-Report	
input	tns:cDRReport
output	tns:cDRReportResponse
fault	tns:negResponse

A one-way operation type is used when a Switching Function sends a message and does not receive a response to the message.

The CSTA-Delivered-Event uses this type of operation. The operation for the CSTA-Delivered-Event is shown below.

```
<operation name="CSTA-Delivered-Event">
  <input message="tns:deliveredEvent"/>
</operation>
```

The following notation is used to describe a WSDL notification operation type:

CSTA-Delivered-event	
input	tns:deliveredEvent

The CSTA-Route-Request service also uses this type of operation. The operation for CSTA-Route-Request is shown below.

```
<operation name="CSTA-Route-Request">
  <input message="tns:routeRequest"/>
</operation>
```

The following notation is another example of a WSDL one-way operation type:

CSTA-Route-Request input tns:routeRequest
--

WSDL 1.1 does not specify a fault message with a one-way operation type, however ISO/IEC 18051 specifies that a negative acknowledgement can be sent for certain services that do not specify a positive acknowledgement. Therefore a separate WSDL operation is specified in this International Standard that provides a negative acknowledgement. It is not required to send a negative acknowledgement in these cases.

```
<operation name="CSTA-Fault-Notification">  
  <input message="tns:negResponse"/>  
</operation>
```

6.4.3 Summary of CSTA Request/Response & Event Models and WSDL Operation Types

The following list shows how the CSTA request/response models and events are represented with WSDL operation types:

- CSTA Services initiated by a WS-CF
 - With ack. – uses Switching Function WSDL request-response operation type.
 - Without ack. – uses Switching Function WSDL one-way operation type.
- CSTA Event Reports sent by a WS-SF
 - Uses Computing Function WSDL one-way operation type.
- CSTA Services initiated by a WS-SF
 - With ack. – Uses Computing Function WSDL request-response operation type.
 - Without ack. – Uses Computing Function WSDL one-way operation type.

Because WSDL 1.1 does not allow a fault element on one-way operation types, *faults for one-way operations are specified as separate WSDL operations*.

6.5 Port Types

A WSDL port type is a collection of WSDL operations. The WSDL port type will be used to bind to one or more specific protocols and transport data formats.

Each port type specified in this International Standard (beginning in Clause 24) corresponds to a profile specified in 2.1.3 of ISO/IEC 18051. A Service Provider may support other operations in addition to the operations specified in the port types.

6.5.1 Example

An example port type with three CSTA WSDL operations follows:

```
<portType name="samplePortType">  
  <operation name="CSTA-Answer-Call">  
    <input message="tns:answerCall"/>  
    <output message="tns:answerCallResponse"/>  
    <fault name="FaultName" message="tns:negResponse"/>  
  </operation>  
</portType>
```

```

</operation>

<operation name="CSTA-Clear-Connection">
  <input message="tns:clearConnection"/>
  <output message="tns:clearConnectionResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Make-Call">
  <input message="tns:makeCall"/>
  <output message="tns:makeCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
</portType>

```

The following notation is used to describe a portType:

porttype **samplePortType**

operations	CSTA-Answer-Call input tns:answerCall output tns:answerCallResponse fault tns:negResponse CSTA-Clear-Connection input tns:clearConnection output tns:clearConnectionResponse fault tns:negResponse CSTA-Make-Call input tns:makeCall output tns:makeCallResponse fault tns:negResponse
source	<pre> <portType name="samplePortType"> <operation name="CSTA-Answer-Call"> <input message="tns:answerCall"/> <output message="tns:answerCallResponse"/> <fault name="FaultName" message="tns:negResponse"/> </operation> <operation name="CSTA-Clear-Connection"> <input message="tns:clearConnection"/> <output message="tns:clearConnectionResponse"/> <fault name="FaultName" message="tns:negResponse"/> </operation> <operation name="CSTA-Make-Call"> <input message="tns:makeCall"/> <output message="tns:makeCallResponse"/> <fault name="FaultName" message="tns:negResponse"/> </operation> </portType> </pre>

6.6 Bindings

A binding associates one concrete protocol and transport data format specification to a port type.

This International Standard allows any type of binding and binding extensions for the CSTA WSDL port types specified in this International Standard.

An example of a SOAP 1.1/HTTP binding is provided in Annex B.

6.7 Port

A port is a single endpoint defined as a combination of a binding and a network address.

An example of how a port is used in a WSDL document is provided in Annex B.

6.8 Service

A service is a collection of related ports.

An example of how a service is used in a WSDL document is provided in Annex B.

7 Capability Exchange Services – Messages and Operations

This Clause specifies the WSDL messages and operations for the CSTA capability exchange services and messages.

7.1 Messages

message **getCSTAFeatures**

parts	parameter
	element csta:GetCSTAFeatures
source	<code><message name="getCSTAFeatures"></code> <code><part name="parameter" element="csta:GetCSTAFeatures"/></code> <code></message></code>

message **getCSTAFeaturesResponse**

parts	parameter
	element csta:GetCSTAFeaturesResponse
source	<code><message name="getCSTAFeaturesResponse"></code> <code><part name="parameter" element="csta:GetCSTAFeaturesResponse"/></code> <code></message></code>

message **getLogicalDeviceInformation**

parts	parameter
	element csta:GetLogicalDeviceInformation
source	<pre><message name="getLogicalDeviceInformation"> <part name="parameter" element="csta:GetLogicalDeviceInformation"/> </message></pre>

message **getLogicalDeviceInformationResponse**

parts	parameter
	element csta:GetLogicalDeviceInformationResponse
source	<pre><message name="getLogicalDeviceInformationResponse"> <part name="parameter" element="csta:GetLogicalDeviceInformationResponse"/> </message></pre>

message **getPhysicalDeviceInformation**

parts	parameter
	element csta:GetPhysicalDeviceInformation
source	<pre><message name="getPhysicalDeviceInformation"> <part name="parameter" element="csta:GetPhysicalDeviceInformation"/> </message></pre>

message **getPhysicalDeviceInformationResponse**

parts	parameter
	element csta:GetPhysicalDeviceInformationResponse
source	<pre><message name="getPhysicalDeviceInformationResponse"> <part name="parameter" element="csta:GetPhysicalDeviceInformationResponse"/> </message></pre>

message **getSwitchingFunctionCaps**

parts	parameter
	element csta:GetSwitchingFunctionCaps
source	<pre><message name="getSwitchingFunctionCaps"> <part name="parameter" element="csta:GetSwitchingFunctionCaps"/> </message></pre>

message **getSwitchingFunctionCapsResponse**

parts	parameter
	element csta:GetSwitchingFunctionCapsResponse
source	<pre><message name="getSwitchingFunctionCapsResponse"> <part name="parameter" element="csta:GetSwitchingFunctionCapsResponse"/> </message></pre>

message **getSwitchingFunctionDevices**

parts	parameter
	element csta:GetSwitchingFunctionDevices
source	<pre><message name="getSwitchingFunctionDevices"> <part name="parameter" element="csta:GetSwitchingFunctionDevices"/> </message></pre>

message **getSwitchingFunctionDevicesResponse**

parts	parameter
	element csta:GetSwitchingFunctionDevicesResponse
source	<pre><message name="getSwitchingFunctionDevicesResponse"> <part name="parameter" element="csta:GetSwitchingFunctionDevicesResponse"/> </message></pre>

message **switchingFunctionDevices**

parts	parameter
	element csta:SwitchingFunctionDevices
source	<pre><message name="switchingFunctionDevices"> <part name="parameter" element="csta:SwitchingFunctionDevices"/> </message></pre>

7.2 Operations

operations	CSTA-Get-CSTA-Features
	input tns:getCSTAFeatures output tns:getCSTAFeaturesResponse fault tns:negResponse
	CSTA-Get-Logical-Device-Information
	input tns:getLogicalDeviceInformation output tns:getLogicalDeviceInformationResponse fault tns:negResponse
	CSTA-Get-Physical-Device-Information
	input tns:getPhysicalDeviceInformation output tns:getPhysicalDeviceInformationResponse fault tns:negResponse
	CSTA-Get-Switching-Function-Capabilities
	input tns:getSwitchingFunctionCaps output tns:getSwitchingFunctionCapsResponse fault tns:negResponse
	CSTA-Get-Switching-Function-Devices
	input tns:getSwitchingFunctionDevices output tns:getSwitchingFunctionDevicesResponse fault tns:negResponse
source	CSTA-Switching-Function-Devices
	output tns:switchingFunctionDevices <pre> <operation name="CSTA-Get-CSTA-Features"> <input message="tns:getCSTAFeatures"/> <output message="tns:getCSTAFeaturesResponse"/> <fault name="FaultName" message="tns:negResponse"/> </operation> <operation name="CSTA-Get-Logical-Device-Information"> <input message="tns:getLogicalDeviceInformation"/> <output message="tns:getLogicalDeviceInformationResponse"/> <fault name="FaultName" message="tns:negResponse"/> </operation> <operation name="CSTA-Get-Physical-Device-Information"> <input message="tns:getPhysicalDeviceInformation"/> <output message="tns:getPhysicalDeviceInformationResponse"/> <fault name="FaultName" message="tns:negResponse"/> </operation> <operation name="CSTA-Get-Switching-Function-Capabilities"> <input message="tns:getSwitchingFunctionCaps"/> <output message="tns:getSwitchingFunctionCapsResponse"/> <fault name="FaultName" message="tns:negResponse"/> </pre>

```

</operation>
<operation name="CSTA-Get-Switching-Function-Devices">
  <input message="tns:getSwitchingFunctionDevices"/>
  <output message="tns:getSwitchingFunctionDevicesResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Switching-Function-Devices">
  <output message="tns:switchingFunctionDevices"/>
</operation>

```

8 System Services – Messages and Operations

This Clause specifies the WSDL messages and operations for the CSTA system services.

8.1 Messages

message **changeSystemStatFilter**

parts	parameter element csta:ChangeSysStatFilter
source	<message name="changeSystemStatFilter"> <part name="parameter" element="csta:ChangeSysStatFilter"/> </message>

message **changeSysStatFilterResponse**

parts	parameter element csta:ChangeSysStatFilterResponse
source	<message name="changeSysStatFilterResponse"> <part name="parameter" element="csta:ChangeSysStatFilterResponse"/> </message>

message **systemRegister**

parts	parameter element csta:SystemRegister
source	<message name="systemRegister"> <part name="parameter" element="csta:SystemRegister"/> </message>

message **systemRegisterResponse**

parts	parameter
	element csta:SystemRegisterResponse
source	<code><message name="systemRegisterResponse"></code> <code><part name="parameter" element="csta:SystemRegisterResponse"/></code> <code></message></code>

message **systemRegisterAbort**

parts	parameter
	element csta:SystemRegisterAbort
source	<code><message name="systemRegisterAbort"></code> <code><part name="parameter" element="csta:SystemRegisterAbort"/></code> <code></message></code>

message **systemRegisterCancel**

parts	parameter
	element csta:SystemRegisterCancel
source	<code><message name="systemRegisterCancel"></code> <code><part name="parameter" element="csta:SystemRegisterCancel"/></code> <code></message></code>

message **systemRegisterCancelResponse**

parts	parameter
	element csta:SystemRegisterCancelResponse
source	<code><message name="systemRegisterCancelResponse"></code> <code><part name="parameter" element="csta:SystemRegisterCancelResponse"/></code> <code></message></code>

message **requestSystemStatus**

parts	parameter
	element csta:RequestSystemStatus
source	<code><message name="requestSystemStatus"></code> <code><part name="parameter" element="csta:RequestSystemStatus"/></code> <code></message></code>

message **requestSystemStatusResponse**

parts	parameter
	element csta:RequestSystemStatusResponse
source	<message name="requestSystemStatusResponse"> <part name="parameter" element="csta:RequestSystemStatusResponse"/> </message>

message **systemStatus**

parts	parameter
	element csta:SystemStatus
source	<message name="systemStatus"> <part name="parameter" element="csta:SystemStatus"/> </message>

message **systemStatusResponse**

parts	parameter
	element csta:SystemStatusResponse
source	<message name="systemStatusResponse"> <part name="parameter" element="csta:SystemStatusResponse"/> </message>

message **swFunctionCapsChanged**

parts	parameter
	element csta:SwFunctionCapsChanged
source	<message name="swFunctionCapsChanged"> <part name="parameter" element="csta:SwFunctionCapsChanged"/> </message>

message **swFunctionCapsChangedResponse**

parts	parameter
	element csta:SwFunctionCapsChangedResponse
source	<message name="swFunctionCapsChangedResponse"> <part name="parameter" element="csta:SwFunctionCapsChangedResponse"/> </message>

message **swFunctionDevicesChanged**

parts	parameter
element	csta:SwFunctionDevicesChanged
source	<pre> <message name="swFunctionDevicesChanged"> <part name="parameter" element="csta:SwFunctionDevicesChanged"/> </message> </pre>

message **swFunctionDevicesChangedResponse**

parts	parameter
element	csta:SwFunctionDevicesChangedResponse
source	<pre> <message name="swFunctionDevicesChangedResponse"> <part name="parameter" element="csta:SwFunctionDevicesChangedResponse"/> </message> </pre>

message **getRegistrations**

parts	parameter
element	csta:GetRegistrations
source	<pre> <message name="getRegistrations"> <part name="parameter" element="csta:GetRegistrations"/> </message> </pre>

message **getRegistrationsResponse**

parts	parameter
element	csta:GetRegistrationsResponse
source	<pre> <message name="getRegistrationsResponse"> <part name="parameter" element="csta:GetRegistrationsResponse"/> </message> </pre>

message **registrationInfo**

parts	parameter
element	csta:RegistrationInfo
source	<pre> <message name="registrationInfo"> <part name="parameter" element="csta:RegistrationInfo"/> </message> </pre>

8.2 Operations

operations

CSTA-Change-System-Status-Filter

input [tns:changeSystemStatFilter](#)
output [tns:changeSysStatFilterResponse](#)
fault [tns:negResponse](#)

CSTA-System-Register

input [tns:systemRegister](#)
output [tns:systemRegisterResponse](#)
fault [tns:negResponse](#)

CSTA-System-Register-Abort

input [tns:systemRegisterAbort](#)

CSTA-System-Register-Cancel

input [tns:systemRegisterCancel](#)
output [tns:systemRegisterCancelResponse](#)
fault [tns:negResponse](#)

CSTA-Request-System-Status-as-input

input [tns:requestSystemStatus](#)
output [tns:requestSystemStatusResponse](#)
fault [tns:negResponse](#)

CSTA-Request-System-Status-as-output

input [tns:requestSystemStatus](#)
output [tns:requestSystemStatusResponse](#)
fault [tns:negResponse](#)

CSTA-System-Status-as-input

input [tns:systemStatus](#)
output [tns:systemStatusResponse](#)
fault [tns:negResponse](#)

CSTA-System-Status-as-output

input [tns:systemStatus](#)
output [tns:systemStatusResponse](#)
fault [tns:negResponse](#)

CSTA-SF-Capabilities-Changed

input [tns:swFunctionCapsChanged](#)
output [tns:swFunctionCapsChangedResponse](#)
fault [tns:negResponse](#)

CSTA-SF-Devices-Changed

input [tns:swFunctionDevicesChanged](#)
 output [tns:swFunctionDevicesChangedResponse](#)
 fault [tns:negResponse](#)

CSTA-Get-Registrations

input **tns:getRegistrations**
 output [tns:getRegistrationsResponse](#)
 fault [tns:negResponse](#)

CSTA-Registration-Info

source input [tns:registrationInfo](#)

```

<operation name="CSTA-Change-System-Status-Filter">
  <input message="tns:changeSystemStatFilter"/>
  <output message="tns:changeSysStatFilterResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-System-Register">
  <input message="tns:systemRegister"/>
  <output message="tns:systemRegisterResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-System-Register-Abort">
  <input message="tns:systemRegisterAbort"/>
</operation>
<operation name="CSTA-System-Register-Cancel">
  <input message="tns:systemRegisterCancel"/>
  <output message="tns:systemRegisterCancelResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Request-System-Status-as-input">
  <input message="tns:requestSystemStatus"/>
  <output message="tns:requestSystemStatusResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Request-System-Status-as-output">
  <input message="tns:requestSystemStatus"/>
  <output message="tns:requestSystemStatusResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-System-Status-as-input">
  <input message="tns:systemStatus"/>
  <output message="tns:systemStatusResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>

```

```

</operation>
<operation name="CSTA-System-Status-as-output">
  <input message="tns:systemStatus"/>
  <output message="tns:systemStatusResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-SF-Capabilities-Changed">
  <input message="tns:swFunctionCapsChanged"/>
  <output message="tns:swFunctionCapsChangedResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-SF-Devices-Changed">
  <input message="tns:swFunctionDevicesChanged"/>
  <output message="tns:swFunctionDevicesChangedResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Registrations">
  <input message="tns:getRegistrations"/>
  <output message="tns:getRegistrationsResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Registration-Info">
  <input message="tns:registrationInfo"/>
</operation>

```

9 Monitoring Services – Messages and Operations

This Clause specifies the WSDL messages and operations for the CSTA monitoring services.

9.1 Messages

message **changeMonitorFilter**

parts	parameter
	element csta:ChangeMonitorFilter
source	<pre> <message name="changeMonitorFilter"> <part name="parameter" element="csta:ChangeMonitorFilter"/> </message> </pre>

message **changeMonitorFilterResponse**

parts	parameter
	element csta:ChangeMonitorFilterResponse
source	<pre> <message name="changeMonitorFilterResponse"> <part name="parameter" element="csta:ChangeMonitorFilterResponse"/> </message> </pre>

message **monitorStart**

parts	parameter
element	csta:MonitorStart
source	<pre><message name="monitorStart"> <part name="parameter" element="csta:MonitorStart"/> </message></pre>

message **monitorStartResponse**

parts	parameter
element	csta:MonitorStartResponse
source	<pre><message name="monitorStartResponse"> <part name="parameter" element="csta:MonitorStartResponse"/> </message></pre>

message **monitorStop**

parts	parameter
element	csta:MonitorStop
source	<pre><message name="monitorStop"> <part name="parameter" element="csta:MonitorStop"/> </message></pre>

message **monitorStopResponse**

parts	parameter
element	csta:MonitorStopResponse
source	<pre><message name="monitorStopResponse"> <part name="parameter" element="csta:MonitorStopResponse"/> </message></pre>

message **getMonitors**

parts	parameter
element	csta:GetMonitors
source	<pre><message name="getMonitors"> <part name="parameter" element="csta:GetMonitors"/> </message></pre>

message **getMonitorsResponse**

parts	parameter
element	csta:GetMonitorsResponse
source	<pre><message name="getMonitorsResponse"> <part name="parameter" element="csta:GetMonitorsResponse"/> </message></pre>

message **monitorInfo**

parts	parameter
element	csta:MonitorInfo
source	<pre><message name="monitorInfo"> <part name="parameter" element="csta:MonitorInfo"/> </message></pre>

9.2 Operations

operations	CSTA-Change-Monitor-Filter input tns:changeMonitorFilter output tns:changeMonitorFilterResponse fault tns:negResponse CSTA-Monitor-Start input tns:monitorStart output tns:monitorStartResponse fault tns:negResponse CSTA-Monitor-Stop-as-input input tns:monitorStop output tns:monitorStopResponse fault tns:negResponse CSTA-Monitor-Stop-as-output input tns:monitorStop output tns:monitorStopResponse fault tns:negResponse CSTA-Get-Monitors input tns:getMonitors output tns:getMonitorsResponse fault tns:negResponse CSTA-Monitor-Info input tns:monitorInfo
------------	---

source	<pre><operation name="CSTA-Change-Monitor-Filter"> <input message="tns:changeMonitorFilter"/> <output message="tns:changeMonitorFilterResponse"/> <fault name="FaultName" message="tns:negResponse"/> </operation> <operation name="CSTA-Monitor-Start"> <input message="tns:monitorStart"/> <output message="tns:monitorStartResponse"/> <fault name="FaultName" message="tns:negResponse"/> </operation> <operation name="CSTA-Monitor-Stop-as-input"> <input message="tns:monitorStop"/> <output message="tns:monitorStopResponse"/> <fault name="FaultName" message="tns:negResponse"/> </operation> <operation name="CSTA-Monitor-Stop-as-output"> <input message="tns:monitorStop"/> <output message="tns:monitorStopResponse"/> <fault name="FaultName" message="tns:negResponse"/> </operation> <operation name="CSTA-Get-Monitors"> <input message="tns:getMonitors"/> <output message="tns:getMonitorsResponse"/> <fault name="FaultName" message="tns:negResponse"/> </operation> <operation name="CSTA-Monitor-Info"> <input message="tns:monitorInfo"/> </operation></pre>
--------	--

10 Snapshot Services – Messages and Operations

This Clause specifies the WSDL messages and operations for the CSTA snapshot services.

10.1 Messages

message **snapshotCall**

parts	<table><tr><td>parameter</td><td></td></tr><tr><td>element</td><td>csta:SnapshotCall</td></tr></table>	parameter		element	csta:SnapshotCall
parameter					
element	csta:SnapshotCall				
source	<pre><message name="snapshotCall"> <part name="parameter" element="csta:SnapshotCall"/> </message></pre>				

message **snapshotCallResponse**

parts	parameter
	element csta:SnapshotCallResponse
source	<code><message name="snapshotCallResponse"></code> <code><part name="parameter" element="csta:SnapshotCallResponse"/></code> <code></message></code>

message **snapshotDevice**

parts	parameter
	element csta:SnapshotDevice
source	<code><message name="snapshotDevice"></code> <code><part name="parameter" element="csta:SnapshotDevice"/></code> <code></message></code>

message **snapshotDeviceResponse**

parts	parameter
	element csta:SnapshotDeviceResponse
source	<code><message name="snapshotDeviceResponse"></code> <code><part name="parameter" element="csta:SnapshotDeviceResponse"/></code> <code></message></code>

message **snapshotCallData**

parts	parameter
	element csta:SnapshotCallData
source	<code><message name="snapshotCallData"></code> <code><part name="parameter" element="csta:SnapshotCallData"/></code> <code></message></code>

message **snapshotDeviceData**

parts	parameter
	element csta:SnapshotDeviceData
source	<code><message name="snapshotDeviceData"></code> <code><part name="parameter" element="csta:SnapshotDeviceData"/></code> <code></message></code>

10.2 Operations

operations	CSTA-Snapshot-Call
	input tns:snapshotCall output tns:snapshotCallResponse fault tns:negResponse
	CSTA-Snapshot-Device
	input tns:snapshotDevice output tns:snapshotDeviceResponse fault tns:negResponse
	CSTA-Snapshot-Call-Data
	input tns:snapshotCallData
source	CSTA-Snapshot-Device-Data
	input tns:snapshotDeviceData <operation name="CSTA-Snapshot-Call"> <input message="tns:snapshotCall"/> <output message="tns:snapshotCallResponse"/> <fault name="FaultName" message="tns:negResponse"/> </operation> <operation name="CSTA-Snapshot-Device"> <input message="tns:snapshotDevice"/> <output message="tns:snapshotDeviceResponse"/> <fault name="FaultName" message="tns:negResponse"/> </operation> <operation name="CSTA-Snapshot-Call-Data"> <input message="tns:snapshotCallData"/> </operation> <operation name="CSTA-Snapshot-Device-Data"> <input message="tns:snapshotDeviceData"/> </operation>

11 Call Control Features – Messages and Operations

This Clause specifies the WSDL messages and operations for the CSTA call control features.

11.1 Messages

message **acceptCall**

parts	parameter
element	csta:AcceptCall
source	<pre><message name="acceptCall"> <part name="parameter" element="csta:AcceptCall"/> </message></pre>

message **acceptCallResponse**

parts	parameter
element	csta:AcceptCallResponse
source	<pre><message name="acceptCallResponse"> <part name="parameter" element="csta:AcceptCallResponse"/> </message></pre>

message **alternateCall**

parts	parameter
element	csta:AlternateCall
source	<pre><message name="alternateCall"> <part name="parameter" element="csta:AlternateCall"/> </message></pre>

message **alternateCallResponse**

parts	parameter
element	csta:AlternateCallResponse
source	<pre><message name="alternateCallResponse"> <part name="parameter" element="csta:AlternateCallResponse"/> </message></pre>

message **answerCall**

parts	parameter
	element csta:AnswerCall
source	<pre><message name="answerCall"> <part name="parameter" element="csta:AnswerCall"/> </message></pre>

message **answerCallResponse**

parts	parameter
	element csta:AnswerCallResponse
source	<pre><message name="answerCallResponse"> <part name="parameter" element="csta:AnswerCallResponse"/> </message></pre>

message **callBack**

parts	parameter
	element csta:CallBack
source	<pre><message name="callBack"> <part name="parameter" element="csta:CallBack"/> </message></pre>

message **callBackResponse**

parts	parameter
	element csta:CallBackResponse
source	<pre><message name="callBackResponse"> <part name="parameter" element="csta:CallBackResponse"/> </message></pre>

message **callBackMessage**

parts	parameter
	element csta:CallBackMessage
source	<pre><message name="callBackMessage"> <part name="parameter" element="csta:CallBackMessage"/> </message></pre>

message **callBackMessageResponse**

parts	parameter
	element csta:CallBackMessageResponse
source	<pre><message name="callBackMessageResponse"> <part name="parameter" element="csta:CallBackMessageResponse"/> </message></pre>

message **camponCall**

parts	parameter
	element csta:CamponCall
source	<pre><message name="camponCall"> <part name="parameter" element="csta:CamponCall"/> </message></pre>

message **camponCallResponse**

parts	parameter
	element csta:CamponCallResponse
source	<pre><message name="camponCallResponse"> <part name="parameter" element="csta:CamponCallResponse"/> </message></pre>

message **clearCall**

parts	parameter
	element csta:ClearCall
source	<pre><message name="clearCall"> <part name="parameter" element="csta:ClearCall"/> </message></pre>

message **clearCallResponse**

parts	parameter
	element csta:ClearCallResponse
source	<pre><message name="clearCallResponse"> <part name="parameter" element="csta:ClearCallResponse"/> </message></pre>

message **clearConnection**

parts	parameter
	element csta:ClearConnection
source	<pre><message name="clearConnection"> <part name="parameter" element="csta:ClearConnection"/> </message></pre>

message **clearConnectionResponse**

parts	parameter
	element csta:ClearConnectionResponse
source	<pre><message name="clearConnectionResponse"> <part name="parameter" element="csta:ClearConnectionResponse"/> </message></pre>

message **conferenceCall**

parts	parameter
	element csta:ConferenceCall
source	<pre><message name="conferenceCall"> <part name="parameter" element="csta:ConferenceCall"/> </message></pre>

message **conferenceCallResponse**

parts	parameter
	element csta:ConferenceCallResponse
source	<pre><message name="conferenceCallResponse"> <part name="parameter" element="csta:ConferenceCallResponse"/> </message></pre>

message **consultationCall**

parts	parameter
	element csta:ConsultationCall
source	<pre><message name="consultationCall"> <part name="parameter" element="csta:ConsultationCall"/> </message></pre>

message **consultationCallResponse**

parts	parameter
	element csta:ConsultationCallResponse
source	<code><message name="consultationCallResponse"></code> <code><part name="parameter" element="csta:ConsultationCallResponse"/></code> <code></message></code>

message **deflectCall**

parts	parameter
	element csta:DeflectCall
source	<code><message name="deflectCall"></code> <code><part name="parameter" element="csta:DeflectCall"/></code> <code></message></code>

message **deflectCallResponse**

parts	parameter
	element csta:DeflectCallResponse
source	<code><message name="deflectCallResponse"></code> <code><part name="parameter" element="csta:DeflectCallResponse"/></code> <code></message></code>

message **dialDigits**

parts	parameter
	element csta:DialDigits
source	<code><message name="dialDigits"></code> <code><part name="parameter" element="csta:DialDigits"/></code> <code></message></code>

message **dialDigitsResponse**

parts	parameter
	element csta:DialDigitsResponse
source	<code><message name="dialDigitsResponse"></code> <code><part name="parameter" element="csta:DialDigitsResponse"/></code> <code></message></code>

message **directedPickupCall**

parts	parameter
	element csta:DirectedPickupCall
source	<pre><message name="directedPickupCall"> <part name="parameter" element="csta:DirectedPickupCall"/> </message></pre>

message **directedPickupCallResponse**

parts	parameter
	element csta:DirectedPickupCallResponse
source	<pre><message name="directedPickupCallResponse"> <part name="parameter" element="csta:DirectedPickupCallResponse"/> </message></pre>

message **groupPickupCall**

parts	parameter
	element csta:GroupPickupCall
source	<pre><message name="groupPickupCall"> <part name="parameter" element="csta:GroupPickupCall"/> </message></pre>

message **groupPickupCallResponse**

parts	parameter
	element csta:GroupPickupCallResponse
source	<pre><message name="groupPickupCallResponse"> <part name="parameter" element="csta:GroupPickupCallResponse"/> </message></pre>

message **holdCall**

parts	parameter
	element csta:HoldCall
source	<pre><message name="holdCall"> <part name="parameter" element="csta:HoldCall"/> </message></pre>

message **holdCallResponse**

parts	parameter
	element csta:HoldCallResponse
source	<code><message name="holdCallResponse"></code> <code><part name="parameter" element="csta:HoldCallResponse"/></code> <code></message></code>

message **intrudeCall**

parts	parameter
	element csta:IntrudeCall
source	<code><message name="intrudeCall"></code> <code><part name="parameter" element="csta:IntrudeCall"/></code> <code></message></code>

message **intrudeCallResponse**

parts	parameter
	element csta:IntrudeCallResponse
source	<code><message name="intrudeCallResponse"></code> <code><part name="parameter" element="csta:IntrudeCallResponse"/></code> <code></message></code>

message **joinCall**

parts	parameter
	element csta:JoinCall
source	<code><message name="joinCall"></code> <code><part name="parameter" element="csta:JoinCall"/></code> <code></message></code>

message **joinCallResponse**

parts	parameter
	element csta:JoinCallResponse
source	<code><message name="joinCallResponse"></code> <code><part name="parameter" element="csta:JoinCallResponse"/></code> <code></message></code>

message **makeCall**

parts	parameter
	element csta:MakeCall
source	<pre><message name="makeCall"> <part name="parameter" element="csta:MakeCall"/> </message></pre>

message **makeCallResponse**

parts	parameter
	element csta:MakeCallResponse
source	<pre><message name="makeCallResponse"> <part name="parameter" element="csta:MakeCallResponse"/> </message></pre>

message **makeConnection**

parts	parameter
	element csta:MakeConnection
source	<pre><message name="makeConnection"> <part name="parameter" element="csta:MakeConnection"/> </message></pre>

message **makeConnectionResponse**

parts	parameter
	element csta:MakeConnectionResponse
source	<pre><message name="makeConnectionResponse"> <part name="parameter" element="csta:MakeConnectionResponse"/> </message></pre>

message **makePredictiveCall**

parts	parameter
	element csta:MakePredictiveCall
source	<pre><message name="makePredictiveCall"> <part name="parameter" element="csta:MakePredictiveCall"/> </message></pre>

message **makePredictiveCallResponse**

parts	parameter
	element csta:MakePredictiveCallResponse
source	<code><message name="makePredictiveCallResponse"></code> <code><part name="parameter" element="csta:MakePredictiveCallResponse"/></code> <code></message></code>

message **parkCall**

parts	parameter
	element csta:ParkCall
source	<code><message name="parkCall"></code> <code><part name="parameter" element="csta:ParkCall"/></code> <code></message></code>

message **parkCallResponse**

parts	parameter
	element csta:ParkCallResponse
source	<code><message name="parkCallResponse"></code> <code><part name="parameter" element="csta:ParkCallResponse"/></code> <code></message></code>

message **reconnectCall**

parts	parameter
	element csta:ReconnectCall
source	<code><message name="reconnectCall"></code> <code><part name="parameter" element="csta:ReconnectCall"/></code> <code></message></code>

message **reconnectCallResponse**

parts	parameter
	element csta:ReconnectCallResponse
source	<code><message name="reconnectCallResponse"></code> <code><part name="parameter" element="csta:ReconnectCallResponse"/></code> <code></message></code>

message **retrieveCall**

parts	parameter
	element csta:RetrieveCall
source	<pre><message name="retrieveCall"> <part name="parameter" element="csta:RetrieveCall"/> </message></pre>

message **retrieveCallResponse**

parts	parameter
	element csta:RetrieveCallResponse
source	<pre><message name="retrieveCallResponse"> <part name="parameter" element="csta:RetrieveCallResponse"/> </message></pre>

message **sendMessage**

parts	parameter
	element csta:SendMessage
source	<pre><message name="sendMessage"> <part name="parameter" element="csta:SendMessage"/> </message></pre>

message **sendMessageResponse**

parts	parameter
	element csta:SendMessageResponse
source	<pre><message name="sendMessageResponse"> <part name="parameter" element="csta:SendMessageResponse"/> </message></pre>

message **singleStepConferenceCall**

parts	parameter
	element csta:SingleStepConferenceCall
source	<pre><message name="singleStepConferenceCall"> <part name="parameter" element="csta:SingleStepConferenceCall"/> </message></pre>

message **singleStepConferenceCallResponse**

parts	parameter
	element csta:SingleStepConferenceCallResponse
source	<code><message name="singleStepConferenceCallResponse"></code> <code><part name="parameter" element="csta:SingleStepConferenceCallResponse"/></code> <code></message></code>

message **singleStepTransferCall**

parts	parameter
	element csta:SingleStepTransferCall
source	<code><message name="singleStepTransferCall"></code> <code><part name="parameter" element="csta:SingleStepTransferCall"/></code> <code></message></code>

message **singleStepTransferCallResponse**

parts	parameter
	element csta:SingleStepTransferCallResponse
source	<code><message name="singleStepTransferCallResponse"></code> <code><part name="parameter" element="csta:SingleStepTransferCallResponse"/></code> <code></message></code>

message **transferCall**

parts	parameter
	element csta:TransferCall
source	<code><message name="transferCall"></code> <code><part name="parameter" element="csta:TransferCall"/></code> <code></message></code>

message **transferCallResponse**

parts	parameter
	element csta:TransferCallResponse
source	<code><message name="transferCallResponse"></code> <code><part name="parameter" element="csta:TransferCallResponse"/></code> <code></message></code>

message **bridgedEvent**

parts	parameter
	element csta:BridgedEvent
source	<pre><message name="bridgedEvent"> <part name="parameter" element="csta:BridgedEvent"/> </message></pre>

message **callClearedEvent**

parts	parameter
	element csta:CallClearedEvent
source	<pre><message name="callClearedEvent"> <part name="parameter" element="csta:CallClearedEvent"/> </message></pre>

message **conferencedEvent**

parts	parameter
	element csta:ConferencedEvent
source	<pre><message name="conferencedEvent"> <part name="parameter" element="csta:ConferencedEvent"/> </message></pre>

message **connectionClearedEvent**

parts	parameter
	element csta:ConnectionClearedEvent
source	<pre><message name="connectionClearedEvent"> <part name="parameter" element="csta:ConnectionClearedEvent"/> </message></pre>

message **deliveredEvent**

parts	parameter
	element csta:DeliveredEvent
source	<pre><message name="deliveredEvent"> <part name="parameter" element="csta:DeliveredEvent"/> </message></pre>

message **digitsDialedEvent**

parts	parameter
	element csta:DigitsDialedEvent
source	<code><message name="digitsDialedEvent"></code> <code><part name="parameter" element="csta:DigitsDialedEvent"/></code> <code></message></code>

message **divertedEvent**

parts	parameter
	element csta:DivertedEvent
source	<code><message name="divertedEvent"></code> <code><part name="parameter" element="csta:DivertedEvent"/></code> <code></message></code>

message **establishedEvent**

parts	parameter
	element csta:EstablishedEvent
source	<code><message name="establishedEvent"></code> <code><part name="parameter" element="csta:EstablishedEvent"/></code> <code></message></code>

message **failedEvent**

parts	parameter
	element csta:FailedEvent
source	<code><message name="failedEvent"></code> <code><part name="parameter" element="csta:FailedEvent"/></code> <code></message></code>

message **heldEvent**

parts	parameter
	element csta:HeldEvent
source	<code><message name="heldEvent"></code> <code><part name="parameter" element="csta:HeldEvent"/></code> <code></message></code>

message **networkCapabilitiesChangedEvent**

parts	parameter
	element csta:NetworkCapabilitiesChangedEvent
source	<pre><message name="networkCapabilitiesChangedEvent"> <part name="parameter" element="csta:NetworkCapabilitiesChangedEvent"/> </message></pre>

message **networkReachedEvent**

parts	parameter
	element csta:NetworkReachedEvent
source	<pre><message name="networkReachedEvent"> <part name="parameter" element="csta:NetworkReachedEvent"/> </message></pre>

message **offeredEvent**

parts	parameter
	element csta:OfferedEvent
source	<pre><message name="offeredEvent"> <part name="parameter" element="csta:OfferedEvent"/> </message></pre>

message **originatedEvent**

parts	parameter
	element csta:OriginatedEvent
source	<pre><message name="originatedEvent"> <part name="parameter" element="csta:OriginatedEvent"/> </message></pre>

message **queuedEvent**

parts	parameter
	element csta:QueuedEvent
source	<pre><message name="queuedEvent"> <part name="parameter" element="csta:QueuedEvent"/> </message></pre>

message **retrievedEvent**

parts	parameter
	element csta:RetrievedEvent
source	<code><message name="retrievedEvent"></code> <code><part name="parameter" element="csta:RetrievedEvent"/></code> <code></message></code>

message **serviceInitiatedEvent**

parts	parameter
	element csta:ServiceInitiatedEvent
source	<code><message name="serviceInitiatedEvent"></code> <code><part name="parameter" element="csta:ServiceInitiatedEvent"/></code> <code></message></code>

message **transferredEvent**

parts	parameter
	element csta:TransferredEvent
source	<code><message name="transferredEvent"></code> <code><part name="parameter" element="csta:TransferredEvent"/></code> <code></message></code>