

---

---

**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*



**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](mailto:copyright@iso.org)  
Web [www.iso.org](http://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

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

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

## 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 forth 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	<b>CSTA-Answer-Call</b> input <b>tns:answerCall</b> output <b>tns:answerCallResponse</b> fault <b>tns:negResponse</b>
	<b>CSTA-Clear-Connection</b> input <b>tns:clearConnection</b> output <b>tns:clearConnectionResponse</b> fault <b>tns:negResponse</b>
	<b>CSTA-Make-Call</b> input <b>tns:makeCall</b> output <b>tns:makeCallResponse</b> fault <b>tns:negResponse</b>
source	<pre> &lt;portType name="samplePortType"&gt;   &lt;operation name="CSTA-Answer-Call"&gt;     &lt;input message="tns:answerCall"/&gt;     &lt;output message="tns:answerCallResponse"/&gt;     &lt;fault name="FaultName" message="tns:negResponse"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Clear-Connection"&gt;     &lt;input message="tns:clearConnection"/&gt;     &lt;output message="tns:clearConnectionResponse"/&gt;     &lt;fault name="FaultName" message="tns:negResponse"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Make-Call"&gt;     &lt;input message="tns:makeCall"/&gt;     &lt;output message="tns:makeCallResponse"/&gt;     &lt;fault name="FaultName" message="tns:negResponse"/&gt;   &lt;/operation&gt; &lt;/portType&gt; </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	<b>parameter</b>
	element <b>csta:GetCSTAFeatures</b>
source	<code>&lt;message name="getCSTAFeatures"&gt;</code>
	<code>&lt;part name="parameter" element="csta:GetCSTAFeatures"/&gt;</code>
	<code>&lt;/message&gt;</code>

message **getCSTAFeaturesResponse**

parts	<b>parameter</b>
	element <b>csta:GetCSTAFeaturesResponse</b>
source	<code>&lt;message name="getCSTAFeaturesResponse"&gt;</code>
	<code>&lt;part name="parameter" element="csta:GetCSTAFeaturesResponse"/&gt;</code>
	<code>&lt;/message&gt;</code>

message **getLogicalDeviceInformation**

parts	<b>parameter</b>
	element <b>csta:GetLogicalDeviceInformation</b>
source	<pre>&lt;message name="getLogicalDeviceInformation"&gt;   &lt;part name="parameter" element="csta:GetLogicalDeviceInformation"/&gt; &lt;/message&gt;</pre>

message **getLogicalDeviceInformationResponse**

parts	<b>parameter</b>
	element <b>csta:GetLogicalDeviceInformationResponse</b>
source	<pre>&lt;message name="getLogicalDeviceInformationResponse"&gt;   &lt;part name="parameter" element="csta:GetLogicalDeviceInformationResponse"/&gt; &lt;/message&gt;</pre>

message **getPhysicalDeviceInformation**

parts	<b>parameter</b>
	element <b>csta:GetPhysicalDeviceInformation</b>
source	<pre>&lt;message name="getPhysicalDeviceInformation"&gt;   &lt;part name="parameter" element="csta:GetPhysicalDeviceInformation"/&gt; &lt;/message&gt;</pre>

message **getPhysicalDeviceInformationResponse**

parts	<b>parameter</b>
	element <b>csta:GetPhysicalDeviceInformationResponse</b>
source	<pre>&lt;message name="getPhysicalDeviceInformationResponse"&gt;   &lt;part name="parameter" element="csta:GetPhysicalDeviceInformationResponse"/&gt; &lt;/message&gt;</pre>

message **getSwitchingFunctionCaps**

parts	<b>parameter</b>
	element <b>csta:GetSwitchingFunctionCaps</b>
source	<pre>&lt;message name="getSwitchingFunctionCaps"&gt;   &lt;part name="parameter" element="csta:GetSwitchingFunctionCaps"/&gt; &lt;/message&gt;</pre>

message **getSwitchingFunctionCapsResponse**

parts	<b>parameter</b>
	element <b>csta:GetSwitchingFunctionCapsResponse</b>
source	<pre> &lt;message name="getSwitchingFunctionCapsResponse"&gt;   &lt;part name="parameter" element="csta:GetSwitchingFunctionCapsResponse"/&gt; &lt;/message&gt; </pre>

message **getSwitchingFunctionDevices**

parts	<b>parameter</b>
	element <b>csta:GetSwitchingFunctionDevices</b>
source	<pre> &lt;message name="getSwitchingFunctionDevices"&gt;   &lt;part name="parameter" element="csta:GetSwitchingFunctionDevices"/&gt; &lt;/message&gt; </pre>

message **getSwitchingFunctionDevicesResponse**

parts	<b>parameter</b>
	element <b>csta:GetSwitchingFunctionDevicesResponse</b>
source	<pre> &lt;message name="getSwitchingFunctionDevicesResponse"&gt;   &lt;part name="parameter" element="csta:GetSwitchingFunctionDevicesResponse"/&gt; &lt;/message&gt; </pre>

message **switchingFunctionDevices**

parts	<b>parameter</b>
	element <b>csta:SwitchingFunctionDevices</b>
source	<pre> &lt;message name="switchingFunctionDevices"&gt;   &lt;part name="parameter" element="csta:SwitchingFunctionDevices"/&gt; &lt;/message&gt; </pre>

## 7.2 Operations

operations	<b>CSTA-Get-CSTA-Features</b>
	input <a href="#">tns:getCSTAFeatures</a> output <a href="#">tns:getCSTAFeaturesResponse</a> fault <a href="#">tns:negResponse</a>
	<b>CSTA-Get-Logical-Device-Information</b>
	input <a href="#">tns:getLogicalDeviceInformation</a> output <a href="#">tns:getLogicalDeviceInformationResponse</a> fault <a href="#">tns:negResponse</a>
	<b>CSTA-Get-Physical-Device-Information</b>
	input <a href="#">tns:getPhysicalDeviceInformation</a> output <a href="#">tns:getPhysicalDeviceInformationResponse</a> fault <a href="#">tns:negResponse</a>
	<b>CSTA-Get-Switching-Function-Capabilities</b>
	input <a href="#">tns:getSwitchingFunctionCaps</a> output <a href="#">tns:getSwitchingFunctionCapsResponse</a> fault <a href="#">tns:negResponse</a>
	<b>CSTA-Get-Switching-Function-Devices</b>
	input <a href="#">tns:getSwitchingFunctionDevices</a> output <a href="#">tns:getSwitchingFunctionDevicesResponse</a> fault <a href="#">tns:negResponse</a>
source	<b>CSTA-Switching-Function-Devices</b>
	output <a href="#">tns:switchingFunctionDevices</a> <pre> &lt;operation name="CSTA-Get-CSTA-Features"&gt;   &lt;input message="tns:getCSTAFeatures"/&gt;   &lt;output message="tns:getCSTAFeaturesResponse"/&gt;   &lt;fault name="FaultName" message="tns:negResponse"/&gt; &lt;/operation&gt; &lt;operation name="CSTA-Get-Logical-Device-Information"&gt;   &lt;input message="tns:getLogicalDeviceInformation"/&gt;   &lt;output message="tns:getLogicalDeviceInformationResponse"/&gt;   &lt;fault name="FaultName" message="tns:negResponse"/&gt; &lt;/operation&gt; &lt;operation name="CSTA-Get-Physical-Device-Information"&gt;   &lt;input message="tns:getPhysicalDeviceInformation"/&gt;   &lt;output message="tns:getPhysicalDeviceInformationResponse"/&gt;   &lt;fault name="FaultName" message="tns:negResponse"/&gt; &lt;/operation&gt; &lt;operation name="CSTA-Get-Switching-Function-Capabilities"&gt;   &lt;input message="tns:getSwitchingFunctionCaps"/&gt;   &lt;output message="tns:getSwitchingFunctionCapsResponse"/&gt;   &lt;fault name="FaultName" message="tns:negResponse"/&gt; </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	<b>parameter</b> element <b>csta:ChangeSysStatFilter</b>
source	<message name="changeSystemStatFilter"> <part name="parameter" element="csta:ChangeSysStatFilter"/> </message>

#### message **changeSysStatFilterResponse**

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

#### message **systemRegister**

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

message **systemRegisterResponse**

parts	<b>parameter</b>
	element <b>csta:SystemRegisterResponse</b>
source	<pre> &lt;message name="systemRegisterResponse"&gt;   &lt;part name="parameter" element="csta:SystemRegisterResponse"/&gt; &lt;/message&gt; </pre>

message **systemRegisterAbort**

parts	<b>parameter</b>
	element <b>csta:SystemRegisterAbort</b>
source	<pre> &lt;message name="systemRegisterAbort"&gt;   &lt;part name="parameter" element="csta:SystemRegisterAbort"/&gt; &lt;/message&gt; </pre>

message **systemRegisterCancel**

parts	<b>parameter</b>
	element <b>csta:SystemRegisterCancel</b>
source	<pre> &lt;message name="systemRegisterCancel"&gt;   &lt;part name="parameter" element="csta:SystemRegisterCancel"/&gt; &lt;/message&gt; </pre>

message **systemRegisterCancelResponse**

parts	<b>parameter</b>
	element <b>csta:SystemRegisterCancelResponse</b>
source	<pre> &lt;message name="systemRegisterCancelResponse"&gt;   &lt;part name="parameter" element="csta:SystemRegisterCancelResponse"/&gt; &lt;/message&gt; </pre>

message **requestSystemStatus**

parts	<b>parameter</b>
	element <b>csta:RequestSystemStatus</b>
source	<pre> &lt;message name="requestSystemStatus"&gt;   &lt;part name="parameter" element="csta:RequestSystemStatus"/&gt; &lt;/message&gt; </pre>

message **requestSystemStatusResponse**

parts	<b>parameter</b>
	element <b>csta:RequestSystemStatusResponse</b>
source	<pre>&lt;message name="requestSystemStatusResponse"&gt;   &lt;part name="parameter" element="csta:RequestSystemStatusResponse"/&gt; &lt;/message&gt;</pre>

message **systemStatus**

parts	<b>parameter</b>
	element <b>csta:SystemStatus</b>
source	<pre>&lt;message name="systemStatus"&gt;   &lt;part name="parameter" element="csta:SystemStatus"/&gt; &lt;/message&gt;</pre>

message **systemStatusResponse**

parts	<b>parameter</b>
	element <b>csta:SystemStatusResponse</b>
source	<pre>&lt;message name="systemStatusResponse"&gt;   &lt;part name="parameter" element="csta:SystemStatusResponse"/&gt; &lt;/message&gt;</pre>

message **swFunctionCapsChanged**

parts	<b>parameter</b>
	element <b>csta:SwFunctionCapsChanged</b>
source	<pre>&lt;message name="swFunctionCapsChanged"&gt;   &lt;part name="parameter" element="csta:SwFunctionCapsChanged"/&gt; &lt;/message&gt;</pre>

message **swFunctionCapsChangedResponse**

parts	<b>parameter</b>
	element <b>csta:SwFunctionCapsChangedResponse</b>
source	<pre>&lt;message name="swFunctionCapsChangedResponse"&gt;   &lt;part name="parameter" element="csta:SwFunctionCapsChangedResponse"/&gt; &lt;/message&gt;</pre>



message **swFunctionDevicesChanged**

parts	<b>parameter</b>
element	<b>csta:SwFunctionDevicesChanged</b>
source	<pre> &lt;message name="swFunctionDevicesChanged"&gt;   &lt;part name="parameter" element="csta:SwFunctionDevicesChanged"/&gt; &lt;/message&gt; </pre>

message **swFunctionDevicesChangedResponse**

parts	<b>parameter</b>
element	<b>csta:SwFunctionDevicesChangedResponse</b>
source	<pre> &lt;message name="swFunctionDevicesChangedResponse"&gt;   &lt;part name="parameter" element="csta:SwFunctionDevicesChangedResponse"/&gt; &lt;/message&gt; </pre>

message **getRegistrations**

parts	<b>parameter</b>
element	<b>csta:GetRegistrations</b>
source	<pre> &lt;message name="getRegistrations"&gt;   &lt;part name="parameter" element="csta:GetRegistrations"/&gt; &lt;/message&gt; </pre>

message **getRegistrationsResponse**

parts	<b>parameter</b>
element	<b>csta:GetRegistrationsResponse</b>
source	<pre> &lt;message name="getRegistrationsResponse"&gt;   &lt;part name="parameter" element="csta:GetRegistrationsResponse"/&gt; &lt;/message&gt; </pre>

message **registrationInfo**

parts	<b>parameter</b>
element	<b>csta:RegistrationInfo</b>
source	<pre> &lt;message name="registrationInfo"&gt;   &lt;part name="parameter" element="csta:RegistrationInfo"/&gt; &lt;/message&gt; </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 <b>csta:ChangeMonitorFilter</b>
source	<pre> &lt;message name="changeMonitorFilter"&gt;   &lt;part name="parameter" element="csta:ChangeMonitorFilter"/&gt; &lt;/message&gt; </pre>

#### message **changeMonitorFilterResponse**

parts	parameter
	element <b>csta:ChangeMonitorFilterResponse</b>
source	<pre> &lt;message name="changeMonitorFilterResponse"&gt;   &lt;part name="parameter" element="csta:ChangeMonitorFilterResponse"/&gt; &lt;/message&gt; </pre>

message **monitorStart**

parts	<b>parameter</b>
	element <b>csta:MonitorStart</b>
source	<pre> &lt;message name="monitorStart"&gt;   &lt;part name="parameter" element="csta:MonitorStart"/&gt; &lt;/message&gt; </pre>

message **monitorStartResponse**

parts	<b>parameter</b>
	element <b>csta:MonitorStartResponse</b>
source	<pre> &lt;message name="monitorStartResponse"&gt;   &lt;part name="parameter" element="csta:MonitorStartResponse"/&gt; &lt;/message&gt; </pre>

message **monitorStop**

parts	<b>parameter</b>
	element <b>csta:MonitorStop</b>
source	<pre> &lt;message name="monitorStop"&gt;   &lt;part name="parameter" element="csta:MonitorStop"/&gt; &lt;/message&gt; </pre>

message **monitorStopResponse**

parts	<b>parameter</b>
	element <b>csta:MonitorStopResponse</b>
source	<pre> &lt;message name="monitorStopResponse"&gt;   &lt;part name="parameter" element="csta:MonitorStopResponse"/&gt; &lt;/message&gt; </pre>

message **getMonitors**

parts	<b>parameter</b>
	element <b>csta:GetMonitors</b>
source	<pre> &lt;message name="getMonitors"&gt;   &lt;part name="parameter" element="csta:GetMonitors"/&gt; &lt;/message&gt; </pre>

message **getMonitorsResponse**

parts	<b>parameter</b>
	element <b>csta:GetMonitorsResponse</b>
source	<code>&lt;message name="getMonitorsResponse"&gt;</code> <code>&lt;part name="parameter" element="csta:GetMonitorsResponse"/&gt;</code> <code>&lt;/message&gt;</code>

message **monitorInfo**

parts	<b>parameter</b>
	element <b>csta:MonitorInfo</b>
source	<code>&lt;message name="monitorInfo"&gt;</code> <code>&lt;part name="parameter" element="csta:MonitorInfo"/&gt;</code> <code>&lt;/message&gt;</code>

## 9.2 Operations

operations	<b>CSTA-Change-Monitor-Filter</b>
	input <a href="#">tns:changeMonitorFilter</a>
	output <a href="#">tns:changeMonitorFilterResponse</a>
	fault <a href="#">tns:negResponse</a>
	<b>CSTA-Monitor-Start</b>
	input <a href="#">tns:monitorStart</a>
	output <a href="#">tns:monitorStartResponse</a>
	fault <a href="#">tns:negResponse</a>
	<b>CSTA-Monitor-Stop-as-input</b>
	input <a href="#">tns:monitorStop</a>
	output <a href="#">tns:monitorStopResponse</a>
	fault <a href="#">tns:negResponse</a>
	<b>CSTA-Monitor-Stop-as-output</b>
	input <a href="#">tns:monitorStop</a>
	output <a href="#">tns:monitorStopResponse</a>
	fault <a href="#">tns:negResponse</a>
	<b>CSTA-Get-Monitors</b>
	input <a href="#">tns:getMonitors</a>
	output <a href="#">tns:getMonitorsResponse</a>
	fault <a href="#">tns:negResponse</a>
	<b>CSTA-Monitor-Info</b>
	input <a href="#">tns:monitorInfo</a>

```

source
<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>

```

## 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	parameter
	element <b>csta:SnapshotCall</b>
source	<pre> &lt;message name="snapshotCall"&gt;   &lt;part name="parameter" element="csta:SnapshotCall"/&gt; &lt;/message&gt; </pre>

message **snapshotCallResponse**

parts	<b>parameter</b>
	element <b>csta:SnapshotCallResponse</b>
source	<code>&lt;message name="snapshotCallResponse"&gt;</code> <code>&lt;part name="parameter" element="csta:SnapshotCallResponse"/&gt;</code> <code>&lt;/message&gt;</code>

message **snapshotDevice**

parts	<b>parameter</b>
	element <b>csta:SnapshotDevice</b>
source	<code>&lt;message name="snapshotDevice"&gt;</code> <code>&lt;part name="parameter" element="csta:SnapshotDevice"/&gt;</code> <code>&lt;/message&gt;</code>

message **snapshotDeviceResponse**

parts	<b>parameter</b>
	element <b>csta:SnapshotDeviceResponse</b>
source	<code>&lt;message name="snapshotDeviceResponse"&gt;</code> <code>&lt;part name="parameter" element="csta:SnapshotDeviceResponse"/&gt;</code> <code>&lt;/message&gt;</code>

message **snapshotCallData**

parts	<b>parameter</b>
	element <b>csta:SnapshotCallData</b>
source	<code>&lt;message name="snapshotCallData"&gt;</code> <code>&lt;part name="parameter" element="csta:SnapshotCallData"/&gt;</code> <code>&lt;/message&gt;</code>

message **snapshotDeviceData**

parts	<b>parameter</b>
	element <b>csta:SnapshotDeviceData</b>
source	<code>&lt;message name="snapshotDeviceData"&gt;</code> <code>&lt;part name="parameter" element="csta:SnapshotDeviceData"/&gt;</code> <code>&lt;/message&gt;</code>



## 10.2 Operations

operations	<b>CSTA-Snapshot-Call</b> input <a href="#">tns:snapshotCall</a> output <a href="#">tns:snapshotCallResponse</a> fault <a href="#">tns:negResponse</a>
	<b>CSTA-Snapshot-Device</b> input <a href="#">tns:snapshotDevice</a> output <a href="#">tns:snapshotDeviceResponse</a> fault <a href="#">tns:negResponse</a>
	<b>CSTA-Snapshot-Call-Data</b> input <a href="#">tns:snapshotCallData</a>
source	<b>CSTA-Snapshot-Device-Data</b> input <a href="#">tns:snapshotDeviceData</a> <pre> &lt;operation name="CSTA-Snapshot-Call"&gt;   &lt;input message="tns:snapshotCall"/&gt;   &lt;output message="tns:snapshotCallResponse"/&gt;   &lt;fault name="FaultName" message="tns:negResponse"/&gt; &lt;/operation&gt; &lt;operation name="CSTA-Snapshot-Device"&gt;   &lt;input message="tns:snapshotDevice"/&gt;   &lt;output message="tns:snapshotDeviceResponse"/&gt;   &lt;fault name="FaultName" message="tns:negResponse"/&gt; &lt;/operation&gt; &lt;operation name="CSTA-Snapshot-Call-Data"&gt;   &lt;input message="tns:snapshotCallData"/&gt; &lt;/operation&gt; &lt;operation name="CSTA-Snapshot-Device-Data"&gt;   &lt;input message="tns:snapshotDeviceData"/&gt; &lt;/operation&gt; </pre>

## 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	<b>parameter</b>
	element <b>csta:AcceptCall</b>
source	<pre>&lt;message name="acceptCall"&gt;   &lt;part name="parameter" element="csta:AcceptCall"/&gt; &lt;/message&gt;</pre>

#### message **acceptCallResponse**

parts	<b>parameter</b>
	element <b>csta:AcceptCallResponse</b>
source	<pre>&lt;message name="acceptCallResponse"&gt;   &lt;part name="parameter" element="csta:AcceptCallResponse"/&gt; &lt;/message&gt;</pre>

#### message **alternateCall**

parts	<b>parameter</b>
	element <b>csta:AlternateCall</b>
source	<pre>&lt;message name="alternateCall"&gt;   &lt;part name="parameter" element="csta:AlternateCall"/&gt; &lt;/message&gt;</pre>

#### message **alternateCallResponse**

parts	<b>parameter</b>
	element <b>csta:AlternateCallResponse</b>
source	<pre>&lt;message name="alternateCallResponse"&gt;   &lt;part name="parameter" element="csta:AlternateCallResponse"/&gt; &lt;/message&gt;</pre>

message **answerCall**

parts	<b>parameter</b>
	element <b>csta:AnswerCall</b>
source	<pre> &lt;message name="answerCall"&gt;   &lt;part name="parameter" element="csta:AnswerCall"/&gt; &lt;/message&gt; </pre>

message **answerCallResponse**

parts	<b>parameter</b>
	element <b>csta:AnswerCallResponse</b>
source	<pre> &lt;message name="answerCallResponse"&gt;   &lt;part name="parameter" element="csta:AnswerCallResponse"/&gt; &lt;/message&gt; </pre>

message **callBack**

parts	<b>parameter</b>
	element <b>csta:CallBack</b>
source	<pre> &lt;message name="callBack"&gt;   &lt;part name="parameter" element="csta:CallBack"/&gt; &lt;/message&gt; </pre>

message **callBackResponse**

parts	<b>parameter</b>
	element <b>csta:CallBackResponse</b>
source	<pre> &lt;message name="callBackResponse"&gt;   &lt;part name="parameter" element="csta:CallBackResponse"/&gt; &lt;/message&gt; </pre>

message **callBackMessage**

parts	<b>parameter</b>
	element <b>csta:CallBackMessage</b>
source	<pre> &lt;message name="callBackMessage"&gt;   &lt;part name="parameter" element="csta:CallBackMessage"/&gt; &lt;/message&gt; </pre>

message **callBackMessageResponse**

parts	<b>parameter</b>
	element <b>csta:CallBackMessageResponse</b>
source	<pre> &lt;message name="callBackMessageResponse"&gt;   &lt;part name="parameter" element="csta:CallBackMessageResponse"/&gt; &lt;/message&gt; </pre>

message **camponCall**

parts	<b>parameter</b>
	element <b>csta:CamponCall</b>
source	<pre> &lt;message name="camponCall"&gt;   &lt;part name="parameter" element="csta:CamponCall"/&gt; &lt;/message&gt; </pre>

message **camponCallResponse**

parts	<b>parameter</b>
	element <b>csta:CamponCallResponse</b>
source	<pre> &lt;message name="camponCallResponse"&gt;   &lt;part name="parameter" element="csta:CamponCallResponse"/&gt; &lt;/message&gt; </pre>

message **clearCall**

parts	<b>parameter</b>
	element <b>csta:ClearCall</b>
source	<pre> &lt;message name="clearCall"&gt;   &lt;part name="parameter" element="csta:ClearCall"/&gt; &lt;/message&gt; </pre>

message **clearCallResponse**

parts	<b>parameter</b>
	element <b>csta:ClearCallResponse</b>
source	<pre> &lt;message name="clearCallResponse"&gt;   &lt;part name="parameter" element="csta:ClearCallResponse"/&gt; &lt;/message&gt; </pre>

**message clearConnection**

parts	<b>parameter</b>
	element <b>csta:ClearConnection</b>
source	<pre>&lt;message name="clearConnection"&gt;   &lt;part name="parameter" element="csta:ClearConnection"/&gt; &lt;/message&gt;</pre>

**message clearConnectionResponse**

parts	<b>parameter</b>
	element <b>csta:ClearConnectionResponse</b>
source	<pre>&lt;message name="clearConnectionResponse"&gt;   &lt;part name="parameter" element="csta:ClearConnectionResponse"/&gt; &lt;/message&gt;</pre>

**message conferenceCall**

parts	<b>parameter</b>
	element <b>csta:ConferenceCall</b>
source	<pre>&lt;message name="conferenceCall"&gt;   &lt;part name="parameter" element="csta:ConferenceCall"/&gt; &lt;/message&gt;</pre>

**message conferenceCallResponse**

parts	<b>parameter</b>
	element <b>csta:ConferenceCallResponse</b>
source	<pre>&lt;message name="conferenceCallResponse"&gt;   &lt;part name="parameter" element="csta:ConferenceCallResponse"/&gt; &lt;/message&gt;</pre>

**message consultationCall**

parts	<b>parameter</b>
	element <b>csta:ConsultationCall</b>
source	<pre>&lt;message name="consultationCall"&gt;   &lt;part name="parameter" element="csta:ConsultationCall"/&gt; &lt;/message&gt;</pre>

message **consultationCallResponse**

parts	<b>parameter</b>
	element <b>csta:ConsultationCallResponse</b>
source	<pre> &lt;message name="consultationCallResponse"&gt;   &lt;part name="parameter" element="csta:ConsultationCallResponse"/&gt; &lt;/message&gt; </pre>

message **deflectCall**

parts	<b>parameter</b>
	element <b>csta:DeflectCall</b>
source	<pre> &lt;message name="deflectCall"&gt;   &lt;part name="parameter" element="csta:DeflectCall"/&gt; &lt;/message&gt; </pre>

message **deflectCallResponse**

parts	<b>parameter</b>
	element <b>csta:DeflectCallResponse</b>
source	<pre> &lt;message name="deflectCallResponse"&gt;   &lt;part name="parameter" element="csta:DeflectCallResponse"/&gt; &lt;/message&gt; </pre>

message **dialDigits**

parts	<b>parameter</b>
	element <b>csta:DialDigits</b>
source	<pre> &lt;message name="dialDigits"&gt;   &lt;part name="parameter" element="csta:DialDigits"/&gt; &lt;/message&gt; </pre>

message **dialDigitsResponse**

parts	<b>parameter</b>
	element <b>csta:DialDigitsResponse</b>
source	<pre> &lt;message name="dialDigitsResponse"&gt;   &lt;part name="parameter" element="csta:DialDigitsResponse"/&gt; &lt;/message&gt; </pre>

**message directedPickupCall**

parts	<b>parameter</b>
	element <b>csta:DirectedPickupCall</b>
source	<pre>&lt;message name="directedPickupCall"&gt;   &lt;part name="parameter" element="csta:DirectedPickupCall"/&gt; &lt;/message&gt;</pre>

**message directedPickupCallResponse**

parts	<b>parameter</b>
	element <b>csta:DirectedPickupCallResponse</b>
source	<pre>&lt;message name="directedPickupCallResponse"&gt;   &lt;part name="parameter" element="csta:DirectedPickupCallResponse"/&gt; &lt;/message&gt;</pre>

**message groupPickupCall**

parts	<b>parameter</b>
	element <b>csta:GroupPickupCall</b>
source	<pre>&lt;message name="groupPickupCall"&gt;   &lt;part name="parameter" element="csta:GroupPickupCall"/&gt; &lt;/message&gt;</pre>

**message groupPickupCallResponse**

parts	<b>parameter</b>
	element <b>csta:GroupPickupCallResponse</b>
source	<pre>&lt;message name="groupPickupCallResponse"&gt;   &lt;part name="parameter" element="csta:GroupPickupCallResponse"/&gt; &lt;/message&gt;</pre>

**message holdCall**

parts	<b>parameter</b>
	element <b>csta:HoldCall</b>
source	<pre>&lt;message name="holdCall"&gt;   &lt;part name="parameter" element="csta:HoldCall"/&gt; &lt;/message&gt;</pre>

#### message holdCallResponse

parts	<b>parameter</b>
	element <b>csta:HoldCallResponse</b>
source	<pre>&lt;message name="holdCallResponse"&gt;   &lt;part name="parameter" element="csta:HoldCallResponse"/&gt; &lt;/message&gt;</pre>

#### message intrudeCall

parts	<b>parameter</b>
	element <b>csta:IntrudeCall</b>
source	<pre>&lt;message name="intrudeCall"&gt;   &lt;part name="parameter" element="csta:IntrudeCall"/&gt; &lt;/message&gt;</pre>

#### message intrudeCallResponse

parts	<b>parameter</b>
	element <b>csta:IntrudeCallResponse</b>
source	<pre>&lt;message name="intrudeCallResponse"&gt;   &lt;part name="parameter" element="csta:IntrudeCallResponse"/&gt; &lt;/message&gt;</pre>

#### message joinCall

parts	<b>parameter</b>
	element <b>csta:JoinCall</b>
source	<pre>&lt;message name="joinCall"&gt;   &lt;part name="parameter" element="csta:JoinCall"/&gt; &lt;/message&gt;</pre>

#### message joinCallResponse

parts	<b>parameter</b>
	element <b>csta:JoinCallResponse</b>
source	<pre>&lt;message name="joinCallResponse"&gt;   &lt;part name="parameter" element="csta:JoinCallResponse"/&gt; &lt;/message&gt;</pre>



message **makeCall**

parts	<b>parameter</b>
	element <b>csta:MakeCall</b>
source	<pre> &lt;message name="makeCall"&gt;   &lt;part name="parameter" element="csta:MakeCall"/&gt; &lt;/message&gt; </pre>

message **makeCallResponse**

parts	<b>parameter</b>
	element <b>csta:MakeCallResponse</b>
source	<pre> &lt;message name="makeCallResponse"&gt;   &lt;part name="parameter" element="csta:MakeCallResponse"/&gt; &lt;/message&gt; </pre>

message **makeConnection**

parts	<b>parameter</b>
	element <b>csta:MakeConnection</b>
source	<pre> &lt;message name="makeConnection"&gt;   &lt;part name="parameter" element="csta:MakeConnection"/&gt; &lt;/message&gt; </pre>

message **makeConnectionResponse**

parts	<b>parameter</b>
	element <b>csta:MakeConnectionResponse</b>
source	<pre> &lt;message name="makeConnectionResponse"&gt;   &lt;part name="parameter" element="csta:MakeConnectionResponse"/&gt; &lt;/message&gt; </pre>

message **makePredictiveCall**

parts	<b>parameter</b>
	element <b>csta:MakePredictiveCall</b>
source	<pre> &lt;message name="makePredictiveCall"&gt;   &lt;part name="parameter" element="csta:MakePredictiveCall"/&gt; &lt;/message&gt; </pre>

message **makePredictiveCallResponse**

parts	<b>parameter</b>
	element <b>csta:MakePredictiveCallResponse</b>
source	<pre>&lt;message name="makePredictiveCallResponse"&gt;   &lt;part name="parameter" element="csta:MakePredictiveCallResponse"/&gt; &lt;/message&gt;</pre>

message **parkCall**

parts	<b>parameter</b>
	element <b>csta:ParkCall</b>
source	<pre>&lt;message name="parkCall"&gt;   &lt;part name="parameter" element="csta:ParkCall"/&gt; &lt;/message&gt;</pre>

message **parkCallResponse**

parts	<b>parameter</b>
	element <b>csta:ParkCallResponse</b>
source	<pre>&lt;message name="parkCallResponse"&gt;   &lt;part name="parameter" element="csta:ParkCallResponse"/&gt; &lt;/message&gt;</pre>

message **reconnectCall**

parts	<b>parameter</b>
	element <b>csta:ReconnectCall</b>
source	<pre>&lt;message name="reconnectCall"&gt;   &lt;part name="parameter" element="csta:ReconnectCall"/&gt; &lt;/message&gt;</pre>

message **reconnectCallResponse**

parts	<b>parameter</b>
	element <b>csta:ReconnectCallResponse</b>
source	<pre>&lt;message name="reconnectCallResponse"&gt;   &lt;part name="parameter" element="csta:ReconnectCallResponse"/&gt; &lt;/message&gt;</pre>

message **retrieveCall**

parts	<b>parameter</b>
	element <b>csta:RetrieveCall</b>
source	<pre> &lt;message name="retrieveCall"&gt;   &lt;part name="parameter" element="csta:RetrieveCall"/&gt; &lt;/message&gt; </pre>

message **retrieveCallResponse**

parts	<b>parameter</b>
	element <b>csta:RetrieveCallResponse</b>
source	<pre> &lt;message name="retrieveCallResponse"&gt;   &lt;part name="parameter" element="csta:RetrieveCallResponse"/&gt; &lt;/message&gt; </pre>

message **sendMessage**

parts	<b>parameter</b>
	element <b>csta:SendMessage</b>
source	<pre> &lt;message name="sendMessage"&gt;   &lt;part name="parameter" element="csta:SendMessage"/&gt; &lt;/message&gt; </pre>

message **sendMessageResponse**

parts	<b>parameter</b>
	element <b>csta:SendMessageResponse</b>
source	<pre> &lt;message name="sendMessageResponse"&gt;   &lt;part name="parameter" element="csta:SendMessageResponse"/&gt; &lt;/message&gt; </pre>

message **singleStepConferenceCall**

parts	<b>parameter</b>
	element <b>csta:SingleStepConferenceCall</b>
source	<pre> &lt;message name="singleStepConferenceCall"&gt;   &lt;part name="parameter" element="csta:SingleStepConferenceCall"/&gt; &lt;/message&gt; </pre>

message **singleStepConferenceCallResponse**

parts	<b>parameter</b>
	element <b>csta:SingleStepConferenceCallResponse</b>
source	<pre>&lt;message name="singleStepConferenceCallResponse"&gt;   &lt;part name="parameter" element="csta:SingleStepConferenceCallResponse"/&gt; &lt;/message&gt;</pre>

message **singleStepTransferCall**

parts	<b>parameter</b>
	element <b>csta:SingleStepTransferCall</b>
source	<pre>&lt;message name="singleStepTransferCall"&gt;   &lt;part name="parameter" element="csta:SingleStepTransferCall"/&gt; &lt;/message&gt;</pre>

message **singleStepTransferCallResponse**

parts	<b>parameter</b>
	element <b>csta:SingleStepTransferCallResponse</b>
source	<pre>&lt;message name="singleStepTransferCallResponse"&gt;   &lt;part name="parameter" element="csta:SingleStepTransferCallResponse"/&gt; &lt;/message&gt;</pre>

message **transferCall**

parts	<b>parameter</b>
	element <b>csta:TransferCall</b>
source	<pre>&lt;message name="transferCall"&gt;   &lt;part name="parameter" element="csta:TransferCall"/&gt; &lt;/message&gt;</pre>

message **transferCallResponse**

parts	<b>parameter</b>
	element <b>csta:TransferCallResponse</b>
source	<pre>&lt;message name="transferCallResponse"&gt;   &lt;part name="parameter" element="csta:TransferCallResponse"/&gt; &lt;/message&gt;</pre>

message **bridgedEvent**

parts	<b>parameter</b>
	element <b>csta:BridgedEvent</b>
source	<pre> &lt;message name="bridgedEvent"&gt;   &lt;part name="parameter" element="csta:BridgedEvent"/&gt; &lt;/message&gt; </pre>

message **callClearedEvent**

parts	<b>parameter</b>
	element <b>csta:CallClearedEvent</b>
source	<pre> &lt;message name="callClearedEvent"&gt;   &lt;part name="parameter" element="csta:CallClearedEvent"/&gt; &lt;/message&gt; </pre>

message **conferencedEvent**

parts	<b>parameter</b>
	element <b>csta:ConferencedEvent</b>
source	<pre> &lt;message name="conferencedEvent"&gt;   &lt;part name="parameter" element="csta:ConferencedEvent"/&gt; &lt;/message&gt; </pre>

message **connectionClearedEvent**

parts	<b>parameter</b>
	element <b>csta:ConnectionClearedEvent</b>
source	<pre> &lt;message name="connectionClearedEvent"&gt;   &lt;part name="parameter" element="csta:ConnectionClearedEvent"/&gt; &lt;/message&gt; </pre>

message **deliveredEvent**

parts	<b>parameter</b>
	element <b>csta:DeliveredEvent</b>
source	<pre> &lt;message name="deliveredEvent"&gt;   &lt;part name="parameter" element="csta:DeliveredEvent"/&gt; &lt;/message&gt; </pre>

message **digitsDialedEvent**

parts	<b>parameter</b>
	element <b>csta:DigitsDialedEvent</b>
source	<pre>&lt;message name="digitsDialedEvent"&gt;   &lt;part name="parameter" element="csta:DigitsDialedEvent"/&gt; &lt;/message&gt;</pre>

message **divertedEvent**

parts	<b>parameter</b>
	element <b>csta:DivertedEvent</b>
source	<pre>&lt;message name="divertedEvent"&gt;   &lt;part name="parameter" element="csta:DivertedEvent"/&gt; &lt;/message&gt;</pre>

message **establishedEvent**

parts	<b>parameter</b>
	element <b>csta:EstablishedEvent</b>
source	<pre>&lt;message name="establishedEvent"&gt;   &lt;part name="parameter" element="csta:EstablishedEvent"/&gt; &lt;/message&gt;</pre>

message **failedEvent**

parts	<b>parameter</b>
	element <b>csta:FailedEvent</b>
source	<pre>&lt;message name="failedEvent"&gt;   &lt;part name="parameter" element="csta:FailedEvent"/&gt; &lt;/message&gt;</pre>

message **heldEvent**

parts	<b>parameter</b>
	element <b>csta:HeldEvent</b>
source	<pre>&lt;message name="heldEvent"&gt;   &lt;part name="parameter" element="csta:HeldEvent"/&gt; &lt;/message&gt;</pre>

message **networkCapabilitiesChangedEvent**

parts	<b>parameter</b>
	element <b>csta:NetworkCapabilitiesChangedEvent</b>
source	<pre>&lt;message name="networkCapabilitiesChangedEvent"&gt;   &lt;part name="parameter" element="csta:NetworkCapabilitiesChangedEvent"/&gt; &lt;/message&gt;</pre>

message **networkReachedEvent**

parts	<b>parameter</b>
	element <b>csta:NetworkReachedEvent</b>
source	<pre>&lt;message name="networkReachedEvent"&gt;   &lt;part name="parameter" element="csta:NetworkReachedEvent"/&gt; &lt;/message&gt;</pre>

message **offeredEvent**

parts	<b>parameter</b>
	element <b>csta:OfferedEvent</b>
source	<pre>&lt;message name="offeredEvent"&gt;   &lt;part name="parameter" element="csta:OfferedEvent"/&gt; &lt;/message&gt;</pre>

message **originatedEvent**

parts	<b>parameter</b>
	element <b>csta:OriginatedEvent</b>
source	<pre>&lt;message name="originatedEvent"&gt;   &lt;part name="parameter" element="csta:OriginatedEvent"/&gt; &lt;/message&gt;</pre>

message **queuedEvent**

parts	<b>parameter</b>
	element <b>csta:QueuedEvent</b>
source	<pre>&lt;message name="queuedEvent"&gt;   &lt;part name="parameter" element="csta:QueuedEvent"/&gt; &lt;/message&gt;</pre>

message **retrievedEvent**

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

message **serviceInitiatedEvent**

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

message **transferredEvent**

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



## 11.2 Operations

operations

### CSTA-Accept-Call

input [tns:acceptCall](#)  
 output [tns:acceptCallResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Alternate-Call

input [tns:alternateCall](#)  
 output [tns:alternateCallResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Answer-Call

input [tns:answerCall](#)  
 output [tns:answerCallResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Call-Back-Call-Related

input [tns:callBack](#)  
 output [tns:callBackResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Call-Back-Message-Call-Related

input [tns:callBackMessage](#)  
 output [tns:callBackMessageResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Camp-On-Call

input [tns:camponCall](#)  
 output [tns:camponCallResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Clear-Call

input [tns:clearCall](#)  
 output [tns:clearCallResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Clear-Connection

input [tns:clearConnection](#)  
 output [tns:clearConnectionResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Conference-Call

input [tns:conferenceCall](#)  
 output [tns:conferenceCallResponse](#)  
 fault [tns:negResponse](#)

#### CSTA-Consultation-Call

input [tns:consultationCall](#)  
output [tns:consultationCallResponse](#)  
fault [tns:negResponse](#)

#### CSTA-Deflect-Call

input [tns:deflectCall](#)  
output [tns:deflectCallResponse](#)  
fault [tns:negResponse](#)

#### CSTA-Dial-Digits-Call

input [tns:dialDigits](#)  
output [tns:dialDigitsResponse](#)  
fault [tns:negResponse](#)

#### CSTA-Directed-Pickup-Call

input [tns:directedPickupCall](#)  
output [tns:directedPickupCallResponse](#)  
fault [tns:negResponse](#)

#### CSTA-Group-Pickup-Call

input [tns:groupPickupCall](#)  
output [tns:groupPickupCallResponse](#)  
fault [tns:negResponse](#)

#### CSTA-Hold-Call

input [tns:holdCall](#)  
output [tns:holdCallResponse](#)  
fault [tns:negResponse](#)

#### CSTA-Intrude-Call

input [tns:intrudeCall](#)  
output [tns:intrudeCallResponse](#)  
fault [tns:negResponse](#)

#### CSTA-Join-Call

input [tns:joinCall](#)  
output [tns:joinCallResponse](#)  
fault [tns:negResponse](#)

#### CSTA-Make-Call

input [tns:makeCall](#)  
output [tns:makeCallResponse](#)  
fault [tns:negResponse](#)

**CSTA-Make-Connection**

input [tns:makeConnection](#)  
 output [tns:makeConnectionResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Park-Call**

input [tns:parkCall](#)  
 output [tns:parkCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Reconnect-Call**

input [tns:reconnectCall](#)  
 output [tns:reconnectCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Retrieve-Call**

input [tns:retrieveCall](#)  
 output [tns:retrieveCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Send-Message**

input [tns:sendMessage](#)  
 output [tns:sendMessageResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Single-Step-Conference-Call**

input [tns:singleStepConferenceCall](#)  
 output [tns:singleStepConferenceCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Single-Step-Transfer-Call**

input [tns:singleStepTransferCall](#)  
 output [tns:singleStepTransferCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Transfer-Call**

input [tns:transferCall](#)  
 output [tns:transferCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Bridged-event**

input [tns:bridgedEvent](#)

**CSTA-Call-Cleared-event**

input [tns:callClearedEvent](#)

**CSTA-Conferenced-event**

input [tns:conferencedEvent](#)

**CSTA-Connection-Cleared-event**

input [tns:connectionClearedEvent](#)

**CSTA-Delivered-event**

input [tns:deliveredEvent](#)

**CSTA-Digits-Dialed-event**

input [tns:digitsDialedEvent](#)

**CSTA-Diverted-event**

input [tns:divertedEvent](#)

**CSTA-Established-event**

input [tns:establishedEvent](#)

**CSTA-Failed-event**

input [tns:failedEvent](#)

**CSTA-Held-event**

input [tns:heldEvent](#)

**CSTA-Network-Capabilities-Changed-event**

input [tns:networkCapabilitiesChangedEvent](#)

**CSTA-Network-Reached-event**

input [tns:networkReachedEvent](#)

**CSTA-Offered-event**

input [tns:offeredEvent](#)

**CSTA-Originated-event**

input [tns:originatedEvent](#)

**CSTA-Queued-event**

input [tns:queuedEvent](#)

**CSTA-Retrieved-event**

input [tns:retrievedEvent](#)

**CSTA-Service-Initiated-event**

input [tns:serviceInitiatedEvent](#)

**CSTA-Transferred-event**

source      input    tns:transferredEvent

```

<operation name="CSTA-Accept-Call">
  <input message="tns:acceptCall"/>
  <output message="tns:acceptCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Alternate-Call">
  <input message="tns:alternateCall"/>
  <output message="tns:alternateCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Answer-Call">
  <input message="tns:answerCall"/>
  <output message="tns:answerCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Call-Back-Call-Related">
  <input message="tns:callBack"/>
  <output message="tns:callBackResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Call-Back-Message-Call-Related">
  <input message="tns:callBackMessage"/>
  <output message="tns:callBackMessageResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Camp-On-Call">
  <input message="tns:camponCall"/>
  <output message="tns:camponCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Clear-Call">
  <input message="tns:clearCall"/>
  <output message="tns:clearCallResponse"/>
  <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-Conference-Call">
  <input message="tns:conferenceCall"/>
  <output message="tns:conferenceCallResponse"/>

```

```

    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Consultation-Call">
    <input message="tns:consultationCall"/>
    <output message="tns:consultationCallResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Deflect-Call">
    <input message="tns:deflectCall"/>
    <output message="tns:deflectCallResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Dial-Digits-Call">
    <input message="tns:dialDigits"/>
    <output message="tns:dialDigitsResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Directed-Pickup-Call">
    <input message="tns:directedPickupCall"/>
    <output message="tns:directedPickupCallResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Group-Pickup-Call">
    <input message="tns:groupPickupCall"/>
    <output message="tns:groupPickupCallResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Hold-Call">
    <input message="tns:holdCall"/>
    <output message="tns:holdCallResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Intrude-Call">
    <input message="tns:intrudeCall"/>
    <output message="tns:intrudeCallResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Join-Call">
    <input message="tns:joinCall"/>
    <output message="tns:joinCallResponse"/>
    <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>
  <operation name="CSTA-Make-Connection">
    <input message="tns:makeConnection"/>
    <output message="tns:makeConnectionResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Make-Predictive-Call">
    <input message="tns:makePredictiveCall"/>
    <output message="tns:makePredictiveCallResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Park-Call">
    <input message="tns:parkCall"/>
    <output message="tns:parkCallResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Reconnect-Call">
    <input message="tns:reconnectCall"/>
    <output message="tns:reconnectCallResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Retrieve-Call">
    <input message="tns:retrieveCall"/>
    <output message="tns:retrieveCallResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Send-Message">
    <input message="tns:sendMessage"/>
    <output message="tns:sendMessageResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Single-Step-Conference-Call">
    <input message="tns:singleStepConferenceCall"/>
    <output message="tns:singleStepConferenceCallResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Single-Step-Transfer-Call">
    <input message="tns:singleStepTransferCall"/>
    <output message="tns:singleStepTransferCallResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Transfer-Call">
    <input message="tns:transferCall"/>
    <output message="tns:transferCallResponse"/>

```

```

    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Bridged-event">
    <input message="tns:bridgedEvent"/>
  </operation>
  <operation name="CSTA-Call-Cleared-event">
    <input message="tns:callClearedEvent"/>
  </operation>
  <operation name="CSTA-Conferenced-event">
    <input message="tns:conferencedEvent"/>
  </operation>
  <operation name="CSTA-Connection-Cleared-event">
    <input message="tns:connectionClearedEvent"/>
  </operation>
  <operation name="CSTA-Delivered-event">
    <input message="tns:deliveredEvent"/>
  </operation>
  <operation name="CSTA-Digits-Dialed-event">
    <input message="tns:digitsDialedEvent"/>
  </operation>
  <operation name="CSTA-Diverted-event">
    <input message="tns:divertedEvent"/>
  </operation>
  <operation name="CSTA-Established-event">
    <input message="tns:establishedEvent"/>
  </operation>
  <operation name="CSTA-Failed-event">
    <input message="tns:failedEvent"/>
  </operation>
  <operation name="CSTA-Held-event">
    <input message="tns:heldEvent"/>
  </operation>
  <operation name="CSTA-Network-Capabilities-Changed-event">
    <input message="tns:networkCapabilitiesChangedEvent"/>
  </operation>
  <operation name="CSTA-Network-Reached-event">
    <input message="tns:networkReachedEvent"/>
  </operation>
  <operation name="CSTA-Offered-event">
    <input message="tns:offeredEvent"/>
  </operation>
  <operation name="CSTA-Originated-event">
    <input message="tns:originatedEvent"/>
  </operation>
  <operation name="CSTA-Queued-event">

```



```

        <input message="tns:queuedEvent"/>
    </operation>
    <operation name="CSTA-Retrieved-event">
        <input message="tns:retrievedEvent"/>
    </operation>
    <operation name="CSTA-Service-Initiated-event">
        <input message="tns:serviceInitiatedEvent"/>
    </operation>
    <operation name="CSTA-Transferred-event">
        <input message="tns:transferredEvent"/>
    </operation>

```

## 12 Call Associated Features - Messages and Operations

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

### 12.1 Messages

message **associateData**

parts	parameter
	element <b>csta:AssociateData</b>
source	<pre> &lt;message name="associateData"&gt;   &lt;part name="parameter" element="csta:AssociateData"/&gt; &lt;/message&gt; </pre>

message **associateDataResponse**

parts	parameter
	element <b>csta:AssociateDataResponse</b>
source	<pre> &lt;message name="associateDataResponse"&gt;   &lt;part name="parameter" element="csta:AssociateDataResponse"/&gt; &lt;/message&gt; </pre>

message **cancelTelephonyTones**

parts	parameter
	element <b>csta:CancelTelephonyTones</b>
source	<pre> &lt;message name="cancelTelephonyTones"&gt;   &lt;part name="parameter" element="csta:CancelTelephonyTones"/&gt; &lt;/message&gt; </pre>

message **cancelTelephonyTonesResponse**

parts	<b>parameter</b>
	element <b>csta:CancelTelephonyTonesResponse</b>
source	<pre>&lt;message name="cancelTelephonyTonesResponse"&gt;   &lt;part name="parameter" element="csta:CancelTelephonyTonesResponse"/&gt; &lt;/message&gt;</pre>

message **changeConnectionInformation**

parts	<b>parameter</b>
	element <b>csta:ChangeConnectionInformation</b>
source	<pre>&lt;message name="changeConnectionInformation"&gt;   &lt;part name="parameter" element="csta:ChangeConnectionInformation"/&gt; &lt;/message&gt;</pre>

message **changeConnectionInformationResponse**

parts	<b>parameter</b>
	element <b>csta:ChangeConnectionInformationResponse</b>
source	<pre>&lt;message name="changeConnectionInformationResponse"&gt;   &lt;part name="parameter" element="csta:ChangeConnectionInformationResponse"/&gt; &lt;/message&gt;</pre>

message **generateDigits**

parts	<b>parameter</b>
	element <b>csta:GenerateDigits</b>
source	<pre>&lt;message name="generateDigits"&gt;   &lt;part name="parameter" element="csta:GenerateDigits"/&gt; &lt;/message&gt;</pre>

message **generateDigitsResponse**

parts	<b>parameter</b>
	element <b>csta:GenerateDigitsResponse</b>
source	<pre>&lt;message name="generateDigitsResponse"&gt;   &lt;part name="parameter" element="csta:GenerateDigitsResponse"/&gt; &lt;/message&gt;</pre>

message **generateTelephonyTones**

parts	<b>parameter</b>
	element <b>csta:GenerateTelephonyTones</b>
source	<pre> &lt;message name="generateTelephonyTones"&gt;   &lt;part name="parameter" element="csta:GenerateTelephonyTones"/&gt; &lt;/message&gt; </pre>

message **generateTelephonyTonesResponse**

parts	<b>parameter</b>
	element <b>csta:GenerateTelephonyTonesResponse</b>
source	<pre> &lt;message name="generateTelephonyTonesResponse"&gt;   &lt;part name="parameter" element="csta:GenerateTelephonyTonesResponse"/&gt; &lt;/message&gt; </pre>

message **sendUserInformation**

parts	<b>parameter</b>
	element <b>csta:SendUserInformation</b>
source	<pre> &lt;message name="sendUserInformation"&gt;   &lt;part name="parameter" element="csta:SendUserInformation"/&gt; &lt;/message&gt; </pre>

message **sendUserInformationResponse**

parts	<b>parameter</b>
	element <b>csta:SendUserInformationResponse</b>
source	<pre> &lt;message name="sendUserInformationResponse"&gt;   &lt;part name="parameter" element="csta:SendUserInformationResponse"/&gt; &lt;/message&gt; </pre>

message **callInformationEvent**

parts	<b>parameter</b>
	element <b>csta:CallInformationEvent</b>
source	<pre> &lt;message name="callInformationEvent"&gt;   &lt;part name="parameter" element="csta:CallInformationEvent"/&gt; &lt;/message&gt; </pre>

message **chargingEvent**

parts	<b>parameter</b>
	element <b>csta:ChargingEvent</b>
source	<pre> &lt;message name="chargingEvent"&gt;   &lt;part name="parameter" element="csta:ChargingEvent"/&gt; &lt;/message&gt; </pre>

message **digitsGeneratedEvent**

parts	<b>parameter</b>
	element <b>csta:DigitsGeneratedEvent</b>
source	<pre> &lt;message name="digitsGeneratedEvent"&gt;   &lt;part name="parameter" element="csta:DigitsGeneratedEvent"/&gt; &lt;/message&gt; </pre>

message **telephonyTonesGeneratedEvent**

parts	<b>parameter</b>
	element <b>csta:TelephonyTonesGeneratedEvent</b>
source	<pre> &lt;message name="telephonyTonesGeneratedEvent"&gt;   &lt;part name="parameter" element="csta:TelephonyTonesGeneratedEvent"/&gt; &lt;/message&gt; </pre>

message **serviceCompletionFailureEvent**

parts	<b>parameter</b>
	element <b>csta:ServiceCompletionFailureEvent</b>
source	<pre> &lt;message name="serviceCompletionFailureEvent"&gt;   &lt;part name="parameter" element="csta:ServiceCompletionFailureEvent"/&gt; &lt;/message&gt; </pre>

## 12.2 Operations

operations

### CSTA-Associate-Data

input [tns:associateData](#)  
 output [tns:associateDataResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Cancel-Telephony-Tones

input [tns:cancelTelephonyTones](#)  
 output [tns:cancelTelephonyTonesResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Change-Connection-Information

input [tns:changeConnectionInformation](#)  
 output [tns:changeConnectionInformationResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Generate-Digits

input [tns:generateDigits](#)  
 output [tns:generateDigitsResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Generate-Telephony-Tones

input [tns:generateTelephonyTones](#)  
 output [tns:generateTelephonyTonesResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Send-User-Information

input [tns:sendUserInformation](#)  
 output [tns:sendUserInformationResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Call-Information-event

input [tns:callInformationEvent](#)

### CSTA-Charging-event

input [tns:chargingEvent](#)

### CSTA-Digits-Generated-event

input [tns:digitsGeneratedEvent](#)

### CSTA-Telephony-Tones-Generated-event

input [tns:telephonyTonesGeneratedEvent](#)

### CSTA-Service-Completion-Failure-event

input [tns:serviceCompletionFailureEvent](#)

source

```

<operation name="CSTA-Associate-Data">
  <input message="tns:associateData"/>
  <output message="tns:associateDataResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Cancel-Telephony-Tones">
  <input message="tns:cancelTelephonyTones"/>
  <output message="tns:cancelTelephonyTonesResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Change-Connection-Information">
  <input message="tns:changeConnectionInformation"/>
  <output message="tns:changeConnectionInformationResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Generate-Digits">
  <input message="tns:generateDigits"/>
  <output message="tns:generateDigitsResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Generate-Telephony-Tones">
  <input message="tns:generateTelephonyTones"/>
  <output message="tns:generateTelephonyTonesResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Send-User-Information">
  <input message="tns:sendUserInfoInformation"/>
  <output message="tns:sendUserInfoInformationResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Call-Information-event">
  <input message="tns:callInformationEvent"/>
</operation>
<operation name="CSTA-Charging-event">
  <input message="tns:chargingEvent"/>
</operation>
<operation name="CSTA-Digits-Generated-event">
  <input message="tns:digitsGeneratedEvent"/>
</operation>
<operation name="CSTA-Telephony-Tones-Generated-event">
  <input message="tns:telephonyTonesGeneratedEvent"/>
</operation>
<operation name="CSTA-Service-Completion-Failure-event">
  <input message="tns:serviceCompletionFailureEvent"/>
</operation>

```

## 13 Media Attachment Features - Messages and Operations

This Clause specifies the WSDL messages and operations for the media attachment features.

### 13.1 Messages

message **attachMediaService**

parts	parameter
	element <b>csta:AttachMediaService</b>
source	<pre>&lt;message name="attachMediaService"&gt;   &lt;part name="parameter" element="csta:AttachMediaService"/&gt; &lt;/message&gt;</pre>

message **attachMediaServiceResponse**

parts	parameter
	element <b>csta:AttachMediaServiceResponse</b>
source	<pre>&lt;message name="attachMediaServiceResponse"&gt;   &lt;part name="parameter" element="csta:AttachMediaServiceResponse"/&gt; &lt;/message&gt;</pre>

message **detachMediaService**

parts	parameter
	element <b>csta:DetachMediaService</b>
source	<pre>&lt;message name="detachMediaService"&gt;   &lt;part name="parameter" element="csta:DetachMediaService"/&gt; &lt;/message&gt;</pre>

message **detachMediaServiceResponse**

parts	parameter
	element <b>csta:DetachMediaServiceResponse</b>
source	<pre>&lt;message name="detachMediaServiceResponse"&gt;   &lt;part name="parameter" element="csta:DetachMediaServiceResponse"/&gt; &lt;/message&gt;</pre>

message **mediaAttachedEvent**

parts	<b>parameter</b>
	element <b>csta:MediaAttachedEvent</b>
source	<pre> &lt;message name="mediaAttachedEvent"&gt;   &lt;part name="parameter" element="csta:MediaAttachedEvent"/&gt; &lt;/message&gt; </pre>

message **mediaDetachedEvent**

parts	<b>parameter</b>
	element <b>csta:MediaDetachedEvent</b>
source	<pre> &lt;message name="mediaDetachedEvent"&gt;   &lt;part name="parameter" element="csta:MediaDetachedEvent"/&gt; &lt;/message&gt; </pre>

## 13.2 Operations

operations	<b>CSTA-Attach-Media-Service</b> input <a href="#">tns:attachMediaService</a> output <a href="#">tns:attachMediaServiceResponse</a> fault <a href="#">tns:negResponse</a>  <b>CSTA-Detach-Media-Service</b> input <a href="#">tns:detachMediaService</a> output <a href="#">tns:detachMediaServiceResponse</a> fault <a href="#">tns:negResponse</a>  <b>CSTA-Media-Attached-Event</b> input <a href="#">tns:mediaAttachedEvent</a>  <b>CSTA-Media-Detached-Event</b> input <a href="#">tns:mediaDetachedEvent</a>
------------	--



```

source
<operation name="CSTA-Attach-Media-Service">
  <input message="tns:attachMediaService"/>
  <output message="tns:attachMediaServiceResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Detach-Media-Service">
  <input message="tns:detachMediaService"/>
  <output message="tns:detachMediaServiceResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Media-Attached-Event">
  <input message="tns:mediaAttachedEvent"/>
</operation>
<operation name="CSTA-Media-Detached-Event">
  <input message="tns:mediaDetachedEvent"/>
</operation>

```

## 14 Routeing Features - Messages and Operations

This Clause specifies the WSDL messages and operations for the routeing features.

### 14.1 Messages

message **routeRegister**

```

parts
  parameter
    element csta:RouteRegister
source
<message name="routeRegister">
  <part name="parameter" element="csta:RouteRegister"/>
</message>

```

message **routeRegisterResponse**

```

parts
  parameter
    element csta:RouteRegisterResponse
source
<message name="routeRegisterResponse">
  <part name="parameter" element="csta:RouteRegisterResponse"/>
</message>

```

message **routeRegisterAbort**

parts	<b>parameter</b>
	element <b>csta:RouteRegisterAbort</b>
source	<pre> &lt;message name="routeRegisterAbort"&gt;   &lt;part name="parameter" element="csta:RouteRegisterAbort"/&gt; &lt;/message&gt; </pre>

message **routeRegisterCancel**

parts	<b>parameter</b>
	element <b>csta:RouteRegisterCancel</b>
source	<pre> &lt;message name="routeRegisterCancel"&gt;   &lt;part name="parameter" element="csta:RouteRegisterCancel"/&gt; &lt;/message&gt; </pre>

message **routeRegisterCancelResponse**

parts	<b>parameter</b>
	element <b>csta:RouteRegisterCancelResponse</b>
source	<pre> &lt;message name="routeRegisterCancelResponse"&gt;   &lt;part name="parameter" element="csta:RouteRegisterCancelResponse"/&gt; &lt;/message&gt; </pre>

message **reRoute**

parts	<b>parameter</b>
	element <b>csta:ReRoute</b>
source	<pre> &lt;message name="reRoute"&gt;   &lt;part name="parameter" element="csta:ReRoute"/&gt; &lt;/message&gt; </pre>

message **routeEnd**

parts	<b>parameter</b>
	element <b>csta:RouteEnd</b>
source	<pre> &lt;message name="routeEnd"&gt;   &lt;part name="parameter" element="csta:RouteEnd"/&gt; &lt;/message&gt; </pre>

message **routeReject**

parts	<b>parameter</b>
	element <b>csta:RouteReject</b>
source	<pre> &lt;message name="routeReject"&gt;   &lt;part name="parameter" element="csta:RouteReject"/&gt; &lt;/message&gt; </pre>

message **routeRequest**

parts	<b>parameter</b>
	element <b>csta:RouteRequest</b>
source	<pre> &lt;message name="routeRequest"&gt;   &lt;part name="parameter" element="csta:RouteRequest"/&gt; &lt;/message&gt; </pre>

message **routeSelect**

parts	<b>parameter</b>
	element <b>csta:RouteSelect</b>
source	<pre> &lt;message name="routeSelect"&gt;   &lt;part name="parameter" element="csta:RouteSelect"/&gt; &lt;/message&gt; </pre>

message **routeUsed**

parts	<b>parameter</b>
	element <b>csta:RouteUsed</b>
source	<pre> &lt;message name="routeUsed"&gt;   &lt;part name="parameter" element="csta:RouteUsed"/&gt; &lt;/message&gt; </pre>

## 14.2 Operations

operations

### CSTA-Route-Register

input [tns:routeRegister](#)  
 output [tns:routeRegisterResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Route-Register-Abort

input [tns:routeRegisterAbort](#)

### CSTA-Route-Register-Cancel

input [tns:routeRegisterCancel](#)  
 output [tns:routeRegisterCancelResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Re-Route

input [tns:reRoute](#)

### CSTA-Route-End-as-input

input [tns:routeEnd](#)

### CSTA-Route-End-as-output

input [tns:routeEnd](#)

### CSTA-Route-Reject

input [tns:routeReject](#)

### CSTA-Route-Request

input [tns:routeRequest](#)

### CSTA-Route-Select

input [tns:routeSelect](#)

### CSTA-Route-Used

input [tns:routeUsed](#)

source

```
<operation name="CSTA-Route-Register">
  <input message="tns:routeRegister"/>
  <output message="tns:routeRegisterResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Route-Register-Abort">
  <input message="tns:routeRegisterAbort"/>
</operation>
<operation name="CSTA-Route-Register-Cancel">
  <input message="tns:routeRegisterCancel"/>
```

```

<output message="tns:routeRegisterCancelResponse"/>
<fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Re-Route">
  <input message="tns:reRoute"/>
</operation>
<operation name="CSTA-Route-End-as-input">
  <input message="tns:routeEnd"/>
</operation>
<operation name="CSTA-Route-End-as-output">
  <input message="tns:routeEnd"/>
</operation>
<operation name="CSTA-Route-Reject">
  <input message="tns:routeReject"/>
</operation>
<operation name="CSTA-Route-Request">
  <input message="tns:routeRequest"/>
</operation>
<operation name="CSTA-Route-Select">
  <input message="tns:routeSelect"/>
</operation>
<operation name="CSTA-Route-Used">
  <input message="tns:routeUsed"/>
</operation>

```

## 15 Physical Device Features - Messages and Operations

This Clause specifies the WSDL messages and operations for the physical device features.

### 15.1 Messages

message **buttonPress**

parts	parameter
	element <b>csta:ButtonPress</b>
source	<pre> &lt;message name="buttonPress"&gt;   &lt;part name="parameter" element="csta:ButtonPress"/&gt; &lt;/message&gt; </pre>

message **buttonPressResponse**

parts	parameter
	element <b>csta:ButtonPressResponse</b>
source	<pre> &lt;message name="buttonPressResponse"&gt;   &lt;part name="parameter" element="csta:ButtonPressResponse"/&gt; &lt;/message&gt; </pre>

message **getAuditoryApparatusInformation**

parts	<b>parameter</b>
	element <b>csta:GetAuditoryApparatusInformation</b>
source	<pre>&lt;message name="getAuditoryApparatusInformation"&gt;   &lt;part name="parameter" element="csta:GetAuditoryApparatusInformation"/&gt; &lt;/message&gt;</pre>

message **getAuditoryApparatusInformationResponse**

parts	<b>parameter</b>
	element <b>csta:GetAuditoryApparatusInformationResponse</b>
source	<pre>&lt;message name="getAuditoryApparatusInformationResponse"&gt;   &lt;part name="parameter" element="csta:GetAuditoryApparatusInformationResponse"/&gt; &lt;/message&gt;</pre>

message **getButtonInformation**

parts	<b>parameter</b>
	element <b>csta:GetButtonInformation</b>
source	<pre>&lt;message name="getButtonInformation"&gt;   &lt;part name="parameter" element="csta:GetButtonInformation"/&gt; &lt;/message&gt;</pre>

message **getButtonInformationResponse**

parts	<b>parameter</b>
	element <b>csta:GetButtonInformationResponse</b>
source	<pre>&lt;message name="getButtonInformationResponse"&gt;   &lt;part name="parameter" element="csta:GetButtonInformationResponse"/&gt; &lt;/message&gt;</pre>

message **getDisplay**

parts	<b>parameter</b>
	element <b>csta:GetDisplay</b>
source	<pre>&lt;message name="getDisplay"&gt;   &lt;part name="parameter" element="csta:GetDisplay"/&gt; &lt;/message&gt;</pre>

**message getDisplayResponse**

parts	<b>parameter</b>
	element <b>csta:GetDisplayResponse</b>
source	<pre>&lt;message name="getDisplayResponse"&gt;   &lt;part name="parameter" element="csta:GetDisplayResponse"/&gt; &lt;/message&gt;</pre>

**message getHookswitchStatus**

parts	<b>parameter</b>
	element <b>csta:GetHookswitchStatus</b>
source	<pre>&lt;message name="getHookswitchStatus"&gt;   &lt;part name="parameter" element="csta:GetHookswitchStatus"/&gt; &lt;/message&gt;</pre>

**message getHookswitchStatusResponse**

parts	<b>parameter</b>
	element <b>csta:GetHookswitchStatusResponse</b>
source	<pre>&lt;message name="getHookswitchStatusResponse"&gt;   &lt;part name="parameter" element="csta:GetHookswitchStatusResponse"/&gt; &lt;/message&gt;</pre>

**message getLampInformation**

parts	<b>parameter</b>
	element <b>csta:GetLampInformation</b>
source	<pre>&lt;message name="getLampInformation"&gt;   &lt;part name="parameter" element="csta:GetLampInformation"/&gt; &lt;/message&gt;</pre>

**message getLampInformationResponse**

parts	<b>parameter</b>
	element <b>csta:GetLampInformationResponse</b>
source	<pre>&lt;message name="getLampInformationResponse"&gt;   &lt;part name="parameter" element="csta:GetLampInformationResponse"/&gt; &lt;/message&gt;</pre>

message **getLampMode**

parts	<b>parameter</b>
	element <b>csta:GetLampMode</b>
source	<message name="getLampMode">
	<part name="parameter" element="csta:GetLampMode"/>
	</message>

message **getLampModeResponse**

parts	<b>parameter</b>
	element <b>csta:GetLampModeResponse</b>
source	<message name="getLampModeResponse">
	<part name="parameter" element="csta:GetLampModeResponse"/>
	</message>

message **getMessageWaitingIndicator**

parts	<b>parameter</b>
	element <b>csta:GetMessageWaitingIndicator</b>
source	<message name="getMessageWaitingIndicator">
	<part name="parameter" element="csta:GetMessageWaitingIndicator"/>
	</message>

message **getMessageWaitingIndicatorResponse**

parts	<b>parameter</b>
	element <b>csta:GetMessageWaitingIndicatorResponse</b>
source	<message name="getMessageWaitingIndicatorResponse">
	<part name="parameter" element="csta:GetMessageWaitingIndicatorResponse"/>
	</message>

message **getMicrophoneGain**

parts	<b>parameter</b>
	element <b>csta:GetMicrophoneGain</b>
source	<message name="getMicrophoneGain">
	<part name="parameter" element="csta:GetMicrophoneGain"/>
	</message>



message **getMicrophoneGainResponse**

parts	<b>parameter</b>
	element <b>csta:GetMicrophoneGainResponse</b>
source	<pre>&lt;message name="getMicrophoneGainResponse"&gt;   &lt;part name="parameter" element="csta:GetMicrophoneGainResponse"/&gt; &lt;/message&gt;</pre>

message **getMicrophoneMute**

parts	<b>parameter</b>
	element <b>csta:GetMicrophoneMute</b>
source	<pre>&lt;message name="getMicrophoneMute"&gt;   &lt;part name="parameter" element="csta:GetMicrophoneMute"/&gt; &lt;/message&gt;</pre>

message **getMicrophoneMuteResponse**

parts	<b>parameter</b>
	element <b>csta:GetMicrophoneMuteResponse</b>
source	<pre>&lt;message name="getMicrophoneMuteResponse"&gt;   &lt;part name="parameter" element="csta:GetMicrophoneMuteResponse"/&gt; &lt;/message&gt;</pre>

message **getRingerStatus**

parts	<b>parameter</b>
	element <b>csta:GetRingerStatus</b>
source	<pre>&lt;message name="getRingerStatus"&gt;   &lt;part name="parameter" element="csta:GetRingerStatus"/&gt; &lt;/message&gt;</pre>

message **getRingerStatusResponse**

parts	<b>parameter</b>
	element <b>csta:GetRingerStatusResponse</b>
source	<pre>&lt;message name="getRingerStatusResponse"&gt;   &lt;part name="parameter" element="csta:GetRingerStatusResponse"/&gt; &lt;/message&gt;</pre>

message **getSpeakerMute**

parts	<b>parameter</b>
	element <b>csta:GetSpeakerMute</b>
source	<message name="getSpeakerMute">
	<part name="parameter" element="csta:GetSpeakerMute"/>
	</message>

message **getSpeakerMuteResponse**

parts	<b>parameter</b>
	element <b>csta:GetSpeakerMuteResponse</b>
source	<message name="getSpeakerMuteResponse">
	<part name="parameter" element="csta:GetSpeakerMuteResponse"/>
	</message>

message **getSpeakerVolume**

parts	<b>parameter</b>
	element <b>csta:GetSpeakerVolume</b>
source	<message name="getSpeakerVolume">
	<part name="parameter" element="csta:GetSpeakerVolume"/>
	</message>

message **getSpeakerVolumeResponse**

parts	<b>parameter</b>
	element <b>csta:GetSpeakerVolumeResponse</b>
source	<message name="getSpeakerVolumeResponse">
	<part name="parameter" element="csta:GetSpeakerVolumeResponse"/>
	</message>

message **setButtonInformation**

parts	<b>parameter</b>
	element <b>csta:SetButtonInformation</b>
source	<message name="setButtonInformation">
	<part name="parameter" element="csta:SetButtonInformation"/>
	</message>

message **setButtonInformationResponse**

parts	<b>parameter</b>
	element <b>csta:SetButtonInformationResponse</b>
source	<pre>&lt;message name="setButtonInformationResponse"&gt;   &lt;part name="parameter" element="csta:SetButtonInformationResponse"/&gt; &lt;/message&gt;</pre>

message **setDisplay**

parts	<b>parameter</b>
	element <b>csta:SetDisplay</b>
source	<pre>&lt;message name="setDisplay"&gt;   &lt;part name="parameter" element="csta:SetDisplay"/&gt; &lt;/message&gt;</pre>

message **setDisplayResponse**

parts	<b>parameter</b>
	element <b>csta:SetDisplayResponse</b>
source	<pre>&lt;message name="setDisplayResponse"&gt;   &lt;part name="parameter" element="csta:SetDisplayResponse"/&gt; &lt;/message&gt;</pre>

message **setHookswitchStatus**

parts	<b>parameter</b>
	element <b>csta:SetHookswitchStatus</b>
source	<pre>&lt;message name="setHookswitchStatus"&gt;   &lt;part name="parameter" element="csta:SetHookswitchStatus"/&gt; &lt;/message&gt;</pre>

message **setHookswitchStatusResponse**

parts	<b>parameter</b>
	element <b>csta:SetHookswitchStatusResponse</b>
source	<pre>&lt;message name="setHookswitchStatusResponse"&gt;   &lt;part name="parameter" element="csta:SetHookswitchStatusResponse"/&gt; &lt;/message&gt;</pre>

message **setLampMode**

parts	<b>parameter</b>
	element <b>csta:SetLampMode</b>
source	<pre>&lt;message name="setLampMode"&gt;   &lt;part name="parameter" element="csta:SetLampMode"/&gt; &lt;/message&gt;</pre>

message **setLampModeResponse**

parts	<b>parameter</b>
	element <b>csta:SetLampModeResponse</b>
source	<pre>&lt;message name="setLampModeResponse"&gt;   &lt;part name="parameter" element="csta:SetLampModeResponse"/&gt; &lt;/message&gt;</pre>

message **setMessageWaitingIndicator**

parts	<b>parameter</b>
	element <b>csta:SetMessageWaitingIndicator</b>
source	<pre>&lt;message name="setMessageWaitingIndicator"&gt;   &lt;part name="parameter" element="csta:SetMessageWaitingIndicator"/&gt; &lt;/message&gt;</pre>

message **setMessageWaitingIndicatorResponse**

parts	<b>parameter</b>
	element <b>csta:SetMessageWaitingIndicatorResponse</b>
source	<pre>&lt;message name="setMessageWaitingIndicatorResponse"&gt;   &lt;part name="parameter" element="csta:SetMessageWaitingIndicatorResponse"/&gt; &lt;/message&gt;</pre>

message **setMicrophoneGain**

parts	<b>parameter</b>
	element <b>csta:SetMicrophoneGain</b>
source	<pre>&lt;message name="setMicrophoneGain"&gt;   &lt;part name="parameter" element="csta:SetMicrophoneGain"/&gt; &lt;/message&gt;</pre>

message **setMicrophoneGainResponse**

parts	<b>parameter</b>
	element <b>csta:SetMicrophoneGainResponse</b>
source	<pre> &lt;message name="setMicrophoneGainResponse"&gt;   &lt;part name="parameter" element="csta:SetMicrophoneGainResponse"/&gt; &lt;/message&gt; </pre>

message **setMicrophoneMute**

parts	<b>parameter</b>
	element <b>csta:SetMicrophoneMute</b>
source	<pre> &lt;message name="setMicrophoneMute"&gt;   &lt;part name="parameter" element="csta:SetMicrophoneMute"/&gt; &lt;/message&gt; </pre>

message **setMicrophoneMuteResponse**

parts	<b>parameter</b>
	element <b>csta:SetMicrophoneMuteResponse</b>
source	<pre> &lt;message name="setMicrophoneMuteResponse"&gt;   &lt;part name="parameter" element="csta:SetMicrophoneMuteResponse"/&gt; &lt;/message&gt; </pre>

message **setRingerStatus**

parts	<b>parameter</b>
	element <b>csta:SetRingerStatus</b>
source	<pre> &lt;message name="setRingerStatus"&gt;   &lt;part name="parameter" element="csta:SetRingerStatus"/&gt; &lt;/message&gt; </pre>

message **setRingerStatusResponse**

parts	<b>parameter</b>
	element <b>csta:SetRingerStatusResponse</b>
source	<pre> &lt;message name="setRingerStatusResponse"&gt;   &lt;part name="parameter" element="csta:SetRingerStatusResponse"/&gt; &lt;/message&gt; </pre>

message **setSpeakerMute**

parts	<b>parameter</b>
	element <b>csta:SetSpeakerMute</b>
source	<message name="setSpeakerMute">
	<part name="parameter" element="csta:SetSpeakerMute"/>
	</message>

message **setSpeakerMuteResponse**

parts	<b>parameter</b>
	element <b>csta:SetSpeakerMuteResponse</b>
source	<message name="setSpeakerMuteResponse">
	<part name="parameter" element="csta:SetSpeakerMuteResponse"/>
	</message>

message **setSpeakerVolume**

parts	<b>parameter</b>
	element <b>csta:SetSpeakerVolume</b>
source	<message name="setSpeakerVolume">
	<part name="parameter" element="csta:SetSpeakerVolume"/>
	</message>

message **setSpeakerVolumeResponse**

parts	<b>parameter</b>
	element <b>csta:SetSpeakerVolumeResponse</b>
source	<message name="setSpeakerVolumeResponse">
	<part name="parameter" element="csta:SetSpeakerVolumeResponse"/>
	</message>

message **buttonInformationEvent**

parts	<b>parameter</b>
	element <b>csta:ButtonInformationEvent</b>
source	<message name="buttonInformationEvent">
	<part name="parameter" element="csta:ButtonInformationEvent"/>
	</message>

message **buttonPressEvent**

parts	<b>parameter</b>
	element <b>csta:ButtonPressEvent</b>
source	<pre> &lt;message name="buttonPressEvent"&gt;   &lt;part name="parameter" element="csta:ButtonPressEvent"/&gt; &lt;/message&gt; </pre>

message **displayUpdatedEvent**

parts	<b>parameter</b>
	element <b>csta:DisplayUpdatedEvent</b>
source	<pre> &lt;message name="displayUpdatedEvent"&gt;   &lt;part name="parameter" element="csta:DisplayUpdatedEvent"/&gt; &lt;/message&gt; </pre>

message **hookswitchEvent**

parts	<b>parameter</b>
	element <b>csta:HookswitchEvent</b>
source	<pre> &lt;message name="hookswitchEvent"&gt;   &lt;part name="parameter" element="csta:HookswitchEvent"/&gt; &lt;/message&gt; </pre>

message **lampModeEvent**

parts	<b>parameter</b>
	element <b>csta:LampModeEvent</b>
source	<pre> &lt;message name="lampModeEvent"&gt;   &lt;part name="parameter" element="csta:LampModeEvent"/&gt; &lt;/message&gt; </pre>

message **messageWaitingEvent**

parts	<b>parameter</b>
	element <b>csta:MessageWaitingEvent</b>
source	<pre> &lt;message name="messageWaitingEvent"&gt;   &lt;part name="parameter" element="csta:MessageWaitingEvent"/&gt; &lt;/message&gt; </pre>

message **microphoneGainEvent**

parts	<b>parameter</b>
	element <b>csta:MicrophoneGainEvent</b>
source	<pre>&lt;message name="microphoneGainEvent"&gt;   &lt;part name="parameter" element="csta:MicrophoneGainEvent"/&gt; &lt;/message&gt;</pre>

message **microphoneMuteEvent**

parts	<b>parameter</b>
	element <b>csta:MicrophoneMuteEvent</b>
source	<pre>&lt;message name="microphoneMuteEvent"&gt;   &lt;part name="parameter" element="csta:MicrophoneMuteEvent"/&gt; &lt;/message&gt;</pre>

message **ringerStatusEvent**

parts	<b>parameter</b>
	element <b>csta:RingerStatusEvent</b>
source	<pre>&lt;message name="ringerStatusEvent"&gt;   &lt;part name="parameter" element="csta:RingerStatusEvent"/&gt; &lt;/message&gt;</pre>

message **speakerMuteEvent**

parts	<b>parameter</b>
	element <b>csta:SpeakerMuteEvent</b>
source	<pre>&lt;message name="speakerMuteEvent"&gt;   &lt;part name="parameter" element="csta:SpeakerMuteEvent"/&gt; &lt;/message&gt;</pre>

message **speakerVolumeEvent**

parts	<b>parameter</b>
	element <b>csta:SpeakerVolumeEvent</b>
source	<pre>&lt;message name="speakerVolumeEvent"&gt;   &lt;part name="parameter" element="csta:SpeakerVolumeEvent"/&gt; &lt;/message&gt;</pre>



## 15.2 Operations

operations

### CSTA-Button-Press

input [tns:buttonPress](#)  
 output [tns:buttonPressResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Get-Auditory-Apparatus-Information

input [tns:getAuditoryApparatusInformation](#)  
 output [tns:getAuditoryApparatusInformationResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Get-Button-Information

input [tns:getButtonInformation](#)  
 output [tns:getButtonInformationResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Get-Display

input [tns:getDisplay](#)  
 output [tns:getDisplayResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Get-Hookswitch-Status

input [tns:getHookswitchStatus](#)  
 output [tns:getHookswitchStatusResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Get-Lamp-Information

input [tns:getLampInformation](#)  
 output [tns:getLampInformationResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Get-Lamp-Mode

input [tns:getLampMode](#)  
 output [tns:getLampModeResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Get-Message-Waiting-Indicator

input [tns:getMessageWaitingIndicator](#)  
 output [tns:getMessageWaitingIndicatorResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Get-Microphone-Gain

input [tns:getMicrophoneGain](#)  
 output [tns:getMicrophoneGainResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Get-Microphone-Mute**

input [tns:getMicrophoneMute](#)  
output [tns:getMicrophoneMuteResponse](#)  
fault [tns:negResponse](#)

**CSTA-Get-Ringer-Status**

input [tns:getRingerStatus](#)  
output [tns:getRingerStatusResponse](#)  
fault [tns:negResponse](#)

**CSTA-Get-Speaker-Mute**

input [tns:getSpeakerMute](#)  
output [tns:getSpeakerMuteResponse](#)  
fault [tns:negResponse](#)

**CSTA-Get-Speaker-Volume**

input [tns:getSpeakerVolume](#)  
output [tns:getSpeakerVolumeResponse](#)  
fault [tns:negResponse](#)

**CSTA-Set-Button-Information**

input [tns:setButtonInformation](#)  
output [tns:setButtonInformationResponse](#)  
fault [tns:negResponse](#)

**CSTA-Set-Display**

input [tns:setDisplay](#)  
output [tns:setDisplayResponse](#)  
fault [tns:negResponse](#)

**CSTA-Set-Hookswitch-Status**

input [tns:setHookswitchStatus](#)  
output [tns:setHookswitchStatusResponse](#)  
fault [tns:negResponse](#)

**CSTA-Set-Lamp-Mode**

input [tns:setLampMode](#)  
output [tns:setLampModeResponse](#)  
fault [tns:negResponse](#)

**CSTA-Set-Message-Waiting-Indicator**

input [tns:setMessageWaitingIndicator](#)  
output [tns:setMessageWaitingIndicatorResponse](#)  
fault [tns:negResponse](#)

**CSTA-Set-Microphone-Gain**

input [tns:setMicrophoneGain](#)  
 output [tns:setMicrophoneGainResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Set-Microphone-Mute**

input [tns:setMicrophoneMute](#)  
 output [tns:setMicrophoneMuteResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Set-Ringer-Status**

input [tns:setRingerStatus](#)  
 output [tns:setRingerStatusResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Set-Speaker-Mute**

input [tns:setSpeakerMute](#)  
 output [tns:setSpeakerMuteResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Set-Speaker-Volume**

input [tns:setSpeakerVolume](#)  
 output [tns:setSpeakerVolumeResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Button-Information-Event**

input [tns:buttonInformationEvent](#)

**CSTA-Button-Press-Event**

input [tns:buttonPressEvent](#)

**CSTA-Display-Updated-Event**

input [tns:displayUpdatedEvent](#)

**CSTA-Hookswitch-Event**

input [tns:hookswitchEvent](#)

**CSTA-Lamp-Mode-Event**

input [tns:lampModeEvent](#)

**CSTA-Message-Waiting-Event**

input [tns:messageWaitingEvent](#)

**CSTA-Microphone-Gain-Event**input [tns:microphoneGainEvent](#)**CSTA-Microphone-Mute-Event**input [tns:microphoneMuteEvent](#)**CSTA-Ringer-Status-Event**input [tns:ringerStatusEvent](#)**CSTA-Speaker-Mute-Event**input [tns:speakerMuteEvent](#)**CSTA-Speaker-Volume-Event**input [tns:speakerVolumeEvent](#)

source

```

<operation name="CSTA-Button-Press">
  <input message="tns:buttonPress"/>
  <output message="tns:buttonPressResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Auditory-Apparatus-Information">
  <input message="tns:getAuditoryApparatusInformation"/>
  <output message="tns:getAuditoryApparatusInformationResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Button-Information">
  <input message="tns:getButtonInformation"/>
  <output message="tns:getButtonInformationResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Display">
  <input message="tns:getDisplay"/>
  <output message="tns:getDisplayResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Hookswitch-Status">
  <input message="tns:getHookswitchStatus"/>
  <output message="tns:getHookswitchStatusResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Lamp-Information">
  <input message="tns:getLampInformation"/>
  <output message="tns:getLampInformationResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>

```

```

</operation>
<operation name="CSTA-Get-Lamp-Mode">
  <input message="tns:getLampMode"/>
  <output message="tns:getLampModeResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Message-Waiting-Indicator">
  <input message="tns:getMessageWaitingIndicator"/>
  <output message="tns:getMessageWaitingIndicatorResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Microphone-Gain">
  <input message="tns:getMicrophoneGain"/>
  <output message="tns:getMicrophoneGainResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Microphone-Mute">
  <input message="tns:getMicrophoneMute"/>
  <output message="tns:getMicrophoneMuteResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Ringer-Status">
  <input message="tns:getRingerStatus"/>
  <output message="tns:getRingerStatusResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Speaker-Mute">
  <input message="tns:getSpeakerMute"/>
  <output message="tns:getSpeakerMuteResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Speaker-Volume">
  <input message="tns:getSpeakerVolume"/>
  <output message="tns:getSpeakerVolumeResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Set-Button-Information">
  <input message="tns:setButtonInformation"/>
  <output message="tns:setButtonInformationResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Set-Display">
  <input message="tns:setDisplay"/>
  <output message="tns:setDisplayResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>

```

```

</operation>
<operation name="CSTA-Set-Hookswitch-Status">
  <input message="tns:setHookswitchStatus"/>
  <output message="tns:setHookswitchStatusResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Set-Lamp-Mode">
  <input message="tns:setLampMode"/>
  <output message="tns:setLampModeResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Set-Message-Waiting-Indicator">
  <input message="tns:setMessageWaitingIndicator"/>
  <output message="tns:setMessageWaitingIndicatorResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Set-Microphone-Gain">
  <input message="tns:setMicrophoneGain"/>
  <output message="tns:setMicrophoneGainResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Set-Microphone-Mute">
  <input message="tns:setMicrophoneMute"/>
  <output message="tns:setMicrophoneMuteResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Set-Ringer-Status">
  <input message="tns:setRingerStatus"/>
  <output message="tns:setRingerStatusResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Set-Speaker-Mute">
  <input message="tns:setSpeakerMute"/>
  <output message="tns:setSpeakerMuteResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Set-Speaker-Volume">
  <input message="tns:setSpeakerVolume"/>
  <output message="tns:setSpeakerVolumeResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Button-Information-Event">
  <input message="tns:buttonInformationEvent"/>
</operation>
<operation name="CSTA-Button-Press-Event">

```

```

    <input message="tns:buttonPressEvent"/>
  </operation>
  <operation name="CSTA-Display-Updated-Event">
    <input message="tns:displayUpdatedEvent"/>
  </operation>
  <operation name="CSTA-Hookswitch-Event">
    <input message="tns:hookswitchEvent"/>
  </operation>
  <operation name="CSTA-Lamp-Mode-Event">
    <input message="tns:lampModeEvent"/>
  </operation>
  <operation name="CSTA-Message-Waiting-Event">
    <input message="tns:messageWaitingEvent"/>
  </operation>
  <operation name="CSTA-Microphone-Gain-Event">
    <input message="tns:microphoneGainEvent"/>
  </operation>
  <operation name="CSTA-Microphone-Mute-Event">
    <input message="tns:microphoneMuteEvent"/>
  </operation>
  <operation name="CSTA-Ringer-Status-Event">
    <input message="tns:ringerStatusEvent"/>
  </operation>
  <operation name="CSTA-Speaker-Mute-Event">
    <input message="tns:speakerMuteEvent"/>
  </operation>
  <operation name="CSTA-Speaker-Volume-Event">
    <input message="tns:speakerVolumeEvent"/>
  </operation>

```

## 16 Logical Device Features - Messages and Operations

This Clause specifies the WSDL messages and operations for the logical device features.

### 16.1 Messages

message **callBackMessageNonCallRelated**

parts	<b>parameter</b>
element	<b>csta:CallBackMessageNonCallRelated</b>
source	<pre> &lt;message name="callBackMessageNonCallRelated"&gt;   &lt;part name="parameter" element="csta:CallBackMessageNonCallRelated"/&gt; &lt;/message&gt; </pre>

message **callBackMessageNonCallRelatedResponse**

parts	<b>parameter</b>
	element <b>csta:CallBackMessageNonCallRelatedResponse</b>
source	<pre> &lt;message name="callBackMessageNonCallRelatedResponse"&gt;   &lt;part name="parameter" element="csta:CallBackMessageNonCallRelatedResponse"/&gt; &lt;/message&gt; </pre>

message **callBackNonCallRelated**

parts	<b>parameter</b>
	element <b>csta:CallBackNonCallRelated</b>
source	<pre> &lt;message name="callBackNonCallRelated"&gt;   &lt;part name="parameter" element="csta:CallBackNonCallRelated"/&gt; &lt;/message&gt; </pre>

message **callBackNonCallRelatedResponse**

parts	<b>parameter</b>
	element <b>csta:CallBackNonCallRelatedResponse</b>
source	<pre> &lt;message name="callBackNonCallRelatedResponse"&gt;   &lt;part name="parameter" element="csta:CallBackNonCallRelatedResponse"/&gt; &lt;/message&gt; </pre>

message **cancelCallBack**

parts	<b>parameter</b>
	element <b>csta:CancelCallBack</b>
source	<pre> &lt;message name="cancelCallBack"&gt;   &lt;part name="parameter" element="csta:CancelCallBack"/&gt; &lt;/message&gt; </pre>

message **cancelCallBackResponse**

parts	<b>parameter</b>
	element <b>csta:CancelCallBackResponse</b>
source	<pre> &lt;message name="cancelCallBackResponse"&gt;   &lt;part name="parameter" element="csta:CancelCallBackResponse"/&gt; &lt;/message&gt; </pre>



message **cancelCallBackMessage**

parts	<b>parameter</b>
	element <b>csta:CancelCallBackMessage</b>
source	<pre>&lt;message name="cancelCallBackMessage"&gt;   &lt;part name="parameter" element="csta:CancelCallBackMessage"/&gt; &lt;/message&gt;</pre>

message **cancelCallBackMessageResponse**

parts	<b>parameter</b>
	element <b>csta:CancelCallBackMessageResponse</b>
source	<pre>&lt;message name="cancelCallBackMessageResponse"&gt;   &lt;part name="parameter" element="csta:CancelCallBackMessageResponse"/&gt; &lt;/message&gt;</pre>

message **getAgentState**

parts	<b>parameter</b>
	element <b>csta:GetAgentState</b>
source	<pre>&lt;message name="getAgentState"&gt;   &lt;part name="parameter" element="csta:GetAgentState"/&gt; &lt;/message&gt;</pre>

message **getAgentStateResponse**

parts	<b>parameter</b>
	element <b>csta:GetAgentStateResponse</b>
source	<pre>&lt;message name="getAgentStateResponse"&gt;   &lt;part name="parameter" element="csta:GetAgentStateResponse"/&gt; &lt;/message&gt;</pre>

message **getAutoAnswer**

parts	<b>parameter</b>
	element <b>csta:GetAutoAnswer</b>
source	<pre>&lt;message name="getAutoAnswer"&gt;   &lt;part name="parameter" element="csta:GetAutoAnswer"/&gt; &lt;/message&gt;</pre>

message **getAutoAnswerResponse**

parts	<b>parameter</b>
	element <b>csta:GetAutoAnswerResponse</b>
source	<message name="getAutoAnswerResponse">
	<part name="parameter" element="csta:GetAutoAnswerResponse"/>
	</message>

message **getAutoWorkMode**

parts	<b>parameter</b>
	element <b>csta:GetAutoWorkMode</b>
source	<message name="getAutoWorkMode">
	<part name="parameter" element="csta:GetAutoWorkMode"/>
	</message>

message **getAutoWorkModeResponse**

parts	<b>parameter</b>
	element <b>csta:GetAutoWorkModeResponse</b>
source	<message name="getAutoWorkModeResponse">
	<part name="parameter" element="csta:GetAutoWorkModeResponse"/>
	</message>

message **getCallBack**

parts	<b>parameter</b>
	element <b>csta:GetCallBack</b>
source	<message name="getCallBack">
	<part name="parameter" element="csta:GetCallBack"/>
	</message>

message **getCallBackResponse**

parts	<b>parameter</b>
	element <b>csta:GetCallBackResponse</b>
source	<message name="getCallBackResponse">
	<part name="parameter" element="csta:GetCallBackResponse"/>
	</message>

message **getCallerIDStatus**

parts	<b>parameter</b>
	element <b>csta:GetCallerIDStatus</b>
source	<pre>&lt;message name="getCallerIDStatus"&gt;   &lt;part name="parameter" element="csta:GetCallerIDStatus"/&gt; &lt;/message&gt;</pre>

message **getCallerIDStatusResponse**

parts	<b>parameter</b>
	element <b>csta:GetCallerIDStatusResponse</b>
source	<pre>&lt;message name="getCallerIDStatusResponse"&gt;   &lt;part name="parameter" element="csta:GetCallerIDStatusResponse"/&gt; &lt;/message&gt;</pre>

message **getDoNotDisturb**

parts	<b>parameter</b>
	element <b>csta:GetDoNotDisturb</b>
source	<pre>&lt;message name="getDoNotDisturb"&gt;   &lt;part name="parameter" element="csta:GetDoNotDisturb"/&gt; &lt;/message&gt;</pre>

message **getDoNotDisturbResponse**

parts	<b>parameter</b>
	element <b>csta:GetDoNotDisturbResponse</b>
source	<pre>&lt;message name="getDoNotDisturbResponse"&gt;   &lt;part name="parameter" element="csta:GetDoNotDisturbResponse"/&gt; &lt;/message&gt;</pre>

message **getForwarding**

parts	<b>parameter</b>
	element <b>csta:GetForwarding</b>
source	<pre>&lt;message name="getForwarding"&gt;   &lt;part name="parameter" element="csta:GetForwarding"/&gt; &lt;/message&gt;</pre>

message **getForwardingResponse**

parts	<b>parameter</b>
	element <b>csta:GetForwardingResponse</b>
source	<pre>&lt;message name="getForwardingResponse"&gt;   &lt;part name="parameter" element="csta:GetForwardingResponse"/&gt; &lt;/message&gt;</pre>

message **getLastNumberDialed**

parts	<b>parameter</b>
	element <b>csta:GetLastNumberDialed</b>
source	<pre>&lt;message name="getLastNumberDialed"&gt;   &lt;part name="parameter" element="csta:GetLastNumberDialed"/&gt; &lt;/message&gt;</pre>

message **getLastNumberDialedResponse**

parts	<b>parameter</b>
	element <b>csta:GetLastNumberDialedResponse</b>
source	<pre>&lt;message name="getLastNumberDialedResponse"&gt;   &lt;part name="parameter" element="csta:GetLastNumberDialedResponse"/&gt; &lt;/message&gt;</pre>

message **getRouteingMode**

parts	<b>parameter</b>
	element <b>csta:GetRouteingMode</b>
source	<pre>&lt;message name="getRouteingMode"&gt;   &lt;part name="parameter" element="csta:GetRouteingMode"/&gt; &lt;/message&gt;</pre>

message **getRouteingModeResponse**

parts	<b>parameter</b>
	element <b>csta:GetRouteingModeResponse</b>
source	<pre>&lt;message name="getRouteingModeResponse"&gt;   &lt;part name="parameter" element="csta:GetRouteingModeResponse"/&gt; &lt;/message&gt;</pre>

message **setAgentState**

parts	<b>parameter</b>
	element <b>csta:SetAgentState</b>
source	<pre>&lt;message name="setAgentState"&gt;   &lt;part name="parameter" element="csta:SetAgentState"/&gt; &lt;/message&gt;</pre>

message **setAgentStateResponse**

parts	<b>parameter</b>
	element <b>csta:SetAgentStateResponse</b>
source	<pre>&lt;message name="setAgentStateResponse"&gt;   &lt;part name="parameter" element="csta:SetAgentStateResponse"/&gt; &lt;/message&gt;</pre>

message **setAutoAnswer**

parts	<b>parameter</b>
	element <b>csta:SetAutoAnswer</b>
source	<pre>&lt;message name="setAutoAnswer"&gt;   &lt;part name="parameter" element="csta:SetAutoAnswer"/&gt; &lt;/message&gt;</pre>

message **setAutoAnswerResponse**

parts	<b>parameter</b>
	element <b>csta:SetAutoAnswerResponse</b>
source	<pre>&lt;message name="setAutoAnswerResponse"&gt;   &lt;part name="parameter" element="csta:SetAutoAnswerResponse"/&gt; &lt;/message&gt;</pre>

message **setAutoWorkMode**

parts	<b>parameter</b>
	element <b>csta:SetAutoWorkMode</b>
source	<pre>&lt;message name="setAutoWorkMode"&gt;   &lt;part name="parameter" element="csta:SetAutoWorkMode"/&gt; &lt;/message&gt;</pre>

message **setAutoWorkModeResponse**

parts	<b>parameter</b>
	element <b>csta:SetAutoWorkModeResponse</b>
source	<message name="setAutoWorkModeResponse"> <part name="parameter" element="csta:SetAutoWorkModeResponse"/> </message>

message **setCallerIDStatus**

parts	<b>parameter</b>
	element <b>csta:SetCallerIDStatus</b>
source	<message name="setCallerIDStatus"> <part name="parameter" element="csta:SetCallerIDStatus"/> </message>

message **setCallerIDStatusResponse**

parts	<b>parameter</b>
	element <b>csta:SetCallerIDStatusResponse</b>
source	<message name="setCallerIDStatusResponse"> <part name="parameter" element="csta:SetCallerIDStatusResponse"/> </message>

message **setDoNotDisturb**

parts	<b>parameter</b>
	element <b>csta:SetDoNotDisturb</b>
source	<message name="setDoNotDisturb"> <part name="parameter" element="csta:SetDoNotDisturb"/> </message>

message **setDoNotDisturbResponse**

parts	<b>parameter</b>
	element <b>csta:SetDoNotDisturbResponse</b>
source	<message name="setDoNotDisturbResponse"> <part name="parameter" element="csta:SetDoNotDisturbResponse"/> </message>

message **setForwarding**

parts	<b>parameter</b>
	element <b>csta:SetForwarding</b>
source	<pre>&lt;message name="setForwarding"&gt;   &lt;part name="parameter" element="csta:SetForwarding"/&gt; &lt;/message&gt;</pre>

message **setForwardingResponse**

parts	<b>parameter</b>
	element <b>csta:SetForwardingResponse</b>
source	<pre>&lt;message name="setForwardingResponse"&gt;   &lt;part name="parameter" element="csta:SetForwardingResponse"/&gt; &lt;/message&gt;</pre>

message **setRouteingMode**

parts	<b>parameter</b>
	element <b>csta:SetRouteingMode</b>
source	<pre>&lt;message name="setRouteingMode"&gt;   &lt;part name="parameter" element="csta:SetRouteingMode"/&gt; &lt;/message&gt;</pre>

message **setRouteingModeResponse**

parts	<b>parameter</b>
	element <b>csta:SetRouteingModeResponse</b>
source	<pre>&lt;message name="setRouteingModeResponse"&gt;   &lt;part name="parameter" element="csta:SetRouteingModeResponse"/&gt; &lt;/message&gt;</pre>

message **agentBusyEvent**

parts	<b>parameter</b>
	element <b>csta:AgentBusyEvent</b>
source	<pre>&lt;message name="agentBusyEvent"&gt;   &lt;part name="parameter" element="csta:AgentBusyEvent"/&gt; &lt;/message&gt;</pre>

message **agentLoggedOffEvent**

parts	<b>parameter</b>
	element <b>csta:AgentLoggedOffEvent</b>
source	<pre> &lt;message name="agentLoggedOffEvent"&gt;   &lt;part name="parameter" element="csta:AgentLoggedOffEvent"/&gt; &lt;/message&gt; </pre>

message **agentLoggedOnEvent**

parts	<b>parameter</b>
	element <b>csta:AgentLoggedOnEvent</b>
source	<pre> &lt;message name="agentLoggedOnEvent"&gt;   &lt;part name="parameter" element="csta:AgentLoggedOnEvent"/&gt; &lt;/message&gt; </pre>

message **agentNotReadyEvent**

parts	<b>parameter</b>
	element <b>csta:AgentNotReadyEvent</b>
source	<pre> &lt;message name="agentNotReadyEvent"&gt;   &lt;part name="parameter" element="csta:AgentNotReadyEvent"/&gt; &lt;/message&gt; </pre>

message **agentReadyEvent**

parts	<b>parameter</b>
	element <b>csta:AgentReadyEvent</b>
source	<pre> &lt;message name="agentReadyEvent"&gt;   &lt;part name="parameter" element="csta:AgentReadyEvent"/&gt; &lt;/message&gt; </pre>

message **agentWorkingAfterCallEvent**

parts	<b>parameter</b>
	element <b>csta:AgentWorkingAfterCallEvent</b>
source	<pre> &lt;message name="agentWorkingAfterCallEvent"&gt;   &lt;part name="parameter" element="csta:AgentWorkingAfterCallEvent"/&gt; &lt;/message&gt; </pre>



message **autoAnswerEvent**

parts	<b>parameter</b>
	element <b>csta:AutoAnswerEvent</b>
source	<pre> &lt;message name="autoAnswerEvent"&gt;   &lt;part name="parameter" element="csta:AutoAnswerEvent"/&gt; &lt;/message&gt; </pre>

message **autoWorkModeEvent**

parts	<b>parameter</b>
	element <b>csta:AutoWorkModeEvent</b>
source	<pre> &lt;message name="autoWorkModeEvent"&gt;   &lt;part name="parameter" element="csta:AutoWorkModeEvent"/&gt; &lt;/message&gt; </pre>

message **callBackEvent**

parts	<b>parameter</b>
	element <b>csta:CallBackEvent</b>
source	<pre> &lt;message name="callBackEvent"&gt;   &lt;part name="parameter" element="csta:CallBackEvent"/&gt; &lt;/message&gt; </pre>

message **callBackMessageEvent**

parts	<b>parameter</b>
	element <b>csta:CallBackMessageEvent</b>
source	<pre> &lt;message name="callBackMessageEvent"&gt;   &lt;part name="parameter" element="csta:CallBackMessageEvent"/&gt; &lt;/message&gt; </pre>

message **callerIDStatusEvent**

parts	<b>parameter</b>
	element <b>csta:CallerIDStatusEvent</b>
source	<pre> &lt;message name="callerIDStatusEvent"&gt;   &lt;part name="parameter" element="csta:CallerIDStatusEvent"/&gt; &lt;/message&gt; </pre>

## message doNotDisturbEvent

parts	<b>parameter</b>
	element <b>csta:DoNotDisturbEvent</b>
source	<pre>&lt;message name="doNotDisturbEvent"&gt;   &lt;part name="parameter" element="csta:DoNotDisturbEvent"/&gt; &lt;/message&gt;</pre>

## message forwardingEvent

parts	<b>parameter</b>
	element <b>csta:ForwardingEvent</b>
source	<pre>&lt;message name="forwardingEvent"&gt;   &lt;part name="parameter" element="csta:ForwardingEvent"/&gt; &lt;/message&gt;</pre>

## message routeingEvent

parts	<b>parameter</b>
	element <b>csta:RouteingEvent</b>
source	<pre>&lt;message name="routeingEvent"&gt;   &lt;part name="parameter" element="csta:RouteingEvent"/&gt; &lt;/message&gt;</pre>

## 16.2 Operations

operations	<b>CSTA-Call-Back-Message-Non-Call-Related</b>
	input <a href="#">tns:callBackMessageNonCallRelated</a>
	output <a href="#">tns:callBackMessageNonCallRelatedResponse</a>
	fault <a href="#">tns:negResponse</a>
	<b>CSTA-Call-Back-Non-Call-Related</b>
	input <a href="#">tns:callBackNonCallRelated</a>
	output <a href="#">tns:callBackNonCallRelatedResponse</a>
	fault <a href="#">tns:negResponse</a>
	<b>CSTA-Cancel-Call-Back</b>
	input <a href="#">tns:cancelCallBack</a>
	output <a href="#">tns:cancelCallBackResponse</a>
	fault <a href="#">tns:negResponse</a>
	<b>CSTA-Cancel-Call-Back-Message</b>
	input <a href="#">tns:cancelCallBackMessage</a>
	output <a href="#">tns:cancelCallBackMessageResponse</a>
	fault <a href="#">tns:negResponse</a>

**CSTA-Get-Agent-State**

input [tns:getAgentState](#)  
 output [tns:getAgentStateResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Get-Auto-Answer**

input [tns:getAutoAnswer](#)  
 output [tns:getAutoAnswerResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Get-Auto-Work-Mode**

input [tns:getAutoWorkMode](#)  
 output [tns:getAutoWorkModeResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Get-Call-Back**

input [tns:getCallBack](#)  
 output [tns:getCallBackResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Get-Caller-ID-Status**

input [tns:getCallerIDStatus](#)  
 output [tns:getCallerIDStatusResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Get-Do-Not-Disturb**

input [tns:getDoNotDisturb](#)  
 output [tns:getDoNotDisturbResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Get-Forwarding**

input [tns:getForwarding](#)  
 output [tns:getForwardingResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Get-Last-Number-Dialed**

input [tns:getLastNumberDialed](#)  
 output [tns:getLastNumberDialedResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Get-Routeing-Mode**

input [tns:getRouteingMode](#)  
 output [tns:getRouteingModeResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Set-Agent-State**

input [tns:setAgentState](#)  
output [tns:setAgentStateResponse](#)  
fault [tns:negResponse](#)

**CSTA-Set-Auto-Answer**

input [tns:setAutoAnswer](#)  
output [tns:setAutoAnswerResponse](#)  
fault [tns:negResponse](#)

**CSTA-Set-Auto-Work-Mode**

input [tns:setAutoWorkMode](#)  
output [tns:setAutoWorkModeResponse](#)  
fault [tns:negResponse](#)

**CSTA-Set-Caller-ID-Status**

input [tns:setCallerIDStatus](#)  
output [tns:setCallerIDStatusResponse](#)  
fault [tns:negResponse](#)

**CSTA-Set-Do-Not-Disturb**

input [tns:setDoNotDisturb](#)  
output [tns:setDoNotDisturbResponse](#)  
fault [tns:negResponse](#)

**CSTA-Set-Forwarding**

input [tns:setForwarding](#)  
output [tns:setForwardingResponse](#)  
fault [tns:negResponse](#)

**CSTA-Set-Routeing-Mode**

input [tns:setRouteingMode](#)  
output [tns:setRouteingModeResponse](#)  
fault [tns:negResponse](#)

**CSTA-Agent-Busy-Event**

input [tns:agentBusyEvent](#)

**CSTA-Agent-Logged-Off-Event**

input [tns:agentLoggedOffEvent](#)

**CSTA-Agent-Logged-On-Event**

input [tns:agentLoggedOnEvent](#)

**CSTA-Agent-Not-Ready-Event**input [tns:agentNotReadyEvent](#)**CSTA-Agent-Ready-Event**input [tns:agentReadyEvent](#)**CSTA-Agent-Working-After-Call-Event**input [tns:agentWorkingAfterCallEvent](#)**CSTA-Auto-Answer-Event**input [tns:autoAnswerEvent](#)**CSTA-Auto-Work-Mode-Event**input [tns:autoWorkModeEvent](#)**CSTA-Call-Back-Event**input [tns:callBackEvent](#)**CSTA-Call-Back-Message-Event**input [tns:callBackMessageEvent](#)**CSTA-Caller-ID-Status-Event**input [tns:callerIDStatusEvent](#)**CSTA-Do-Not-Disturb-Event**input [tns:doNotDisturbEvent](#)**CSTA-Forwarding-Event**input [tns:forwardingEvent](#)**CSTA-Routeing-Event**input [tns:routeingEvent](#)

source

```

<operation name="CSTA-Call-Back-Message-Non-Call-Related">
  <input message="tns:callBackMessageNonCallRelated"/>
  <output message="tns:callBackMessageNonCallRelatedResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Call-Back-Non-Call-Related">
  <input message="tns:callBackNonCallRelated"/>
  <output message="tns:callBackNonCallRelatedResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

```

```

<operation name="CSTA-Cancel-Call-Back">
  <input message="tns:cancelCallBack"/>
  <output message="tns:cancelCallBackResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Cancel-Call-Back-Message">
  <input message="tns:cancelCallBackMessage"/>
  <output message="tns:cancelCallBackMessageResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Agent-State">
  <input message="tns:getAgentState"/>
  <output message="tns:getAgentStateResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Auto-Answer">
  <input message="tns:getAutoAnswer"/>
  <output message="tns:getAutoAnswerResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Auto-Work-Mode">
  <input message="tns:getAutoWorkMode"/>
  <output message="tns:getAutoWorkModeResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Call-Back">
  <input message="tns:getCallBack"/>
  <output message="tns:getCallBackResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Caller-ID-Status">
  <input message="tns:getCallerIDStatus"/>
  <output message="tns:getCallerIDStatusResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Do-Not-Disturb">
  <input message="tns:getDoNotDisturb"/>
  <output message="tns:getDoNotDisturbResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Forwarding">
  <input message="tns:getForwarding"/>
  <output message="tns:getForwardingResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

```

```

<operation name="CSTA-Get-Last-Number-Dialed">
  <input message="tns:getLastNumberDialed"/>
  <output message="tns:getLastNumberDialedResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Get-Routeing-Mode">
  <input message="tns:getRouteingMode"/>
  <output message="tns:getRouteingModeResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Set-Agent-State">
  <input message="tns:setAgentState"/>
  <output message="tns:setAgentStateResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Set-Auto-Answer">
  <input message="tns:setAutoAnswer"/>
  <output message="tns:setAutoAnswerResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Set-Auto-Work-Mode">
  <input message="tns:setAutoWorkMode"/>
  <output message="tns:setAutoWorkModeResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Set-Caller-ID-Status">
  <input message="tns:setCallerIDStatus"/>
  <output message="tns:setCallerIDStatusResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Set-Do-Not-Disturb">
  <input message="tns:setDoNotDisturb"/>
  <output message="tns:setDoNotDisturbResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Set-Forwarding">
  <input message="tns:setForwarding"/>
  <output message="tns:setForwardingResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Set-Routeing-Mode">
  <input message="tns:setRouteingMode"/>
  <output message="tns:setRouteingModeResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

```

```

<operation name="CSTA-Agent-Busy-Event">
  <input message="tns:agentBusyEvent"/>
</operation>

<operation name="CSTA-Agent-Logged-Off-Event">
  <input message="tns:agentLoggedOffEvent"/>
</operation>

<operation name="CSTA-Agent-Logged-On-Event">
  <input message="tns:agentLoggedOnEvent"/>
</operation>

<operation name="CSTA-Agent-Not-Ready-Event">
  <input message="tns:agentNotReadyEvent"/>
</operation>

<operation name="CSTA-Agent-Ready-Event">
  <input message="tns:agentReadyEvent"/>
</operation>

<operation name="CSTA-Agent-Working-After-Call-Event">
  <input message="tns:agentWorkingAfterCallEvent"/>
</operation>

<operation name="CSTA-Auto-Answer-Event">
  <input message="tns:autoAnswerEvent"/>
</operation>

<operation name="CSTA-Auto-Work-Mode-Event">
  <input message="tns:autoWorkModeEvent"/>
</operation>

<operation name="CSTA-Call-Back-Event">
  <input message="tns:callBackEvent"/>
</operation>

<operation name="CSTA-Call-Back-Message-Event">
  <input message="tns:callBackMessageEvent"/>
</operation>

<operation name="CSTA-Caller-ID-Status-Event">
  <input message="tns:callerIDStatusEvent"/>
</operation>

<operation name="CSTA-Do-Not-Disturb-Event">
  <input message="tns:doNotDisturbEvent"/>
</operation>

<operation name="CSTA-Forwarding-Event">
  <input message="tns:forwardingEvent"/>
</operation>

<operation name="CSTA-Routeing-Event">
  <input message="tns:routeingEvent"/>
</operation>

```



## 17 Device Maintenance Events - Messages and Operations

This Clause specifies the WSDL messages and operations for the device maintenance events.

### 17.1 Messages

message **backInServiceEvent**

parts	<b>parameter</b> element <b>csta:BackInServiceEvent</b>
source	<pre>&lt;message name="backInServiceEvent"&gt;   &lt;part name="parameter" element="csta:BackInServiceEvent"/&gt; &lt;/message&gt;</pre>

message **deviceCapsChangedEvent**

parts	<b>parameter</b> element <b>csta:DeviceCapsChangedEvent</b>
source	<pre>&lt;message name="deviceCapsChangedEvent"&gt;   &lt;part name="parameter" element="csta:DeviceCapsChangedEvent"/&gt; &lt;/message&gt;</pre>

message **outOfServiceEvent**

parts	<b>parameter</b> element <b>csta:OutOfServiceEvent</b>
source	<pre>&lt;message name="outOfServiceEvent"&gt;   &lt;part name="parameter" element="csta:OutOfServiceEvent"/&gt; &lt;/message&gt;</pre>

message **partiallyInServiceEvent**

parts	<b>parameter</b> element <b>csta:PartiallyInServiceEvent</b>
source	<pre>&lt;message name="partiallyInServiceEvent"&gt;   &lt;part name="parameter" element="csta:PartiallyInServiceEvent"/&gt; &lt;/message&gt;</pre>

## 17.2 Operations

operations	<b>CSTA-Back-In-Service-Event</b> input <a href="#">tns:backInServiceEvent</a>  <b>CSTA-Device-Capabilities-Changed-Event</b> input <a href="#">tns:deviceCapsChangedEvent</a>  <b>CSTA-Out-Of-Service-Event</b> input <a href="#">tns:outOfServiceEvent</a>  <b>CSTA-Partially-In-Service-Event</b> input <a href="#">tns:partiallyInServiceEvent</a>
source	<pre> &lt;operation name="CSTA-Back-In-Service-Event"&gt;   &lt;input message="tns:backInServiceEvent"/&gt; &lt;/operation&gt; &lt;operation name="CSTA-Device-Capabilities-Changed-Event"&gt;   &lt;input message="tns:deviceCapsChangedEvent"/&gt; &lt;/operation&gt; &lt;operation name="CSTA-Out-Of-Service-Event"&gt;   &lt;input message="tns:outOfServiceEvent"/&gt; &lt;/operation&gt; &lt;operation name="CSTA-Partially-In-Service-Event"&gt;   &lt;input message="tns:partiallyInServiceEvent"/&gt; &lt;/operation&gt; </pre>

## 18 I/O Services Features - Messages and Operations

This Clause specifies the WSDL messages and operations for the I/O Services features.

### 18.1 Messages

message **iORegister**

parts	<b>parameter</b> element <b>csta:IORegister</b>
source	<pre> &lt;message name="iORegister"&gt;   &lt;part name="parameter" element="csta:IORegister"/&gt; &lt;/message&gt; </pre>

message **iORegisterResponse**

parts	<b>parameter</b> element <b>csta:IORegisterResponse</b>
source	<pre> &lt;message name="iORegisterResponse"&gt;   &lt;part name="parameter" element="csta:IORegisterResponse"/&gt; &lt;/message&gt; </pre>

**message iORegisterAbort**

parts	<b>parameter</b>
	element <b>csta:IORegisterAbort</b>
source	<pre> &lt;message name="iORegisterAbort"&gt;   &lt;part name="parameter" element="csta:IORegisterAbort"/&gt; &lt;/message&gt; </pre>

**message iORegisterCancel**

parts	<b>parameter</b>
	element <b>csta:IORegisterCancel</b>
source	<pre> &lt;message name="iORegisterCancel"&gt;   &lt;part name="parameter" element="csta:IORegisterCancel"/&gt; &lt;/message&gt; </pre>

**message iORegisterCancelResponse**

parts	<b>parameter</b>
	element <b>csta:IORegisterCancelResponse</b>
source	<pre> &lt;message name="iORegisterCancelResponse"&gt;   &lt;part name="parameter" element="csta:IORegisterCancelResponse"/&gt; &lt;/message&gt; </pre>

**message dataPathResumed**

parts	<b>parameter</b>
	element <b>csta:DataPathResumed</b>
source	<pre> &lt;message name="dataPathResumed"&gt;   &lt;part name="parameter" element="csta:DataPathResumed"/&gt; &lt;/message&gt; </pre>

**message dataPathResumedResponse**

parts	<b>parameter</b>
	element <b>csta:DataPathResumedResponse</b>
source	<pre> &lt;message name="dataPathResumedResponse"&gt;   &lt;part name="parameter" element="csta:DataPathResumedResponse"/&gt; &lt;/message&gt; </pre>

message **dataPathSuspended**

parts	<b>parameter</b>
	element <b>csta:DataPathSuspended</b>
source	<pre> &lt;message name="dataPathSuspended"&gt;   &lt;part name="parameter" element="csta:DataPathSuspended"/&gt; &lt;/message&gt; </pre>

message **dataPathSuspendedResponse**

parts	<b>parameter</b>
	element <b>csta:DataPathSuspendedResponse</b>
source	<pre> &lt;message name="dataPathSuspendedResponse"&gt;   &lt;part name="parameter" element="csta:DataPathSuspendedResponse"/&gt; &lt;/message&gt; </pre>

message **fastData**

parts	<b>parameter</b>
	element <b>csta:FastData</b>
source	<pre> &lt;message name="fastData"&gt;   &lt;part name="parameter" element="csta:FastData"/&gt; &lt;/message&gt; </pre>

message **fastDataResponse**

parts	<b>parameter</b>
	element <b>csta:FastDataResponse</b>
source	<pre> &lt;message name="fastDataResponse"&gt;   &lt;part name="parameter" element="csta:FastDataResponse"/&gt; &lt;/message&gt; </pre>

message **resumeDataPath**

parts	<b>parameter</b>
	element <b>csta:ResumeDataPath</b>
source	<pre> &lt;message name="resumeDataPath"&gt;   &lt;part name="parameter" element="csta:ResumeDataPath"/&gt; &lt;/message&gt; </pre>

message **resumeDataPathResponse**

parts	<b>parameter</b>
	element <b>csta:ResumeDataPathResponse</b>
source	<pre> &lt;message name="resumeDataPathResponse"&gt;   &lt;part name="parameter" element="csta:ResumeDataPathResponse"/&gt; &lt;/message&gt; </pre>

message **sendBroadcastData**

parts	<b>parameter</b>
	element <b>csta:SendBroadcastData</b>
source	<pre> &lt;message name="sendBroadcastData"&gt;   &lt;part name="parameter" element="csta:SendBroadcastData"/&gt; &lt;/message&gt; </pre>

message **sendBroadcastDataResponse**

parts	<b>parameter</b>
	element <b>csta:SendBroadcastDataResponse</b>
source	<pre> &lt;message name="sendBroadcastDataResponse"&gt;   &lt;part name="parameter" element="csta:SendBroadcastDataResponse"/&gt; &lt;/message&gt; </pre>

message **sendData**

parts	<b>parameter</b>
	element <b>csta:SendData</b>
source	<pre> &lt;message name="sendData"&gt;   &lt;part name="parameter" element="csta:SendData"/&gt; &lt;/message&gt; </pre>

message **sendDataResponse**

parts	<b>parameter</b>
	element <b>csta:SendDataResponse</b>
source	<pre> &lt;message name="sendDataResponse"&gt;   &lt;part name="parameter" element="csta:SendDataResponse"/&gt; &lt;/message&gt; </pre>

message **sendMulticastData**

parts	<b>parameter</b>
	element <b>csta:SendMulticastData</b>
source	<pre>&lt;message name="sendMulticastData"&gt;   &lt;part name="parameter" element="csta:SendMulticastData"/&gt; &lt;/message&gt;</pre>

message **sendMulticastDataResponse**

parts	<b>parameter</b>
	element <b>csta:SendMulticastDataResponse</b>
source	<pre>&lt;message name="sendMulticastDataResponse"&gt;   &lt;part name="parameter" element="csta:SendMulticastDataResponse"/&gt; &lt;/message&gt;</pre>

message **startDataPath**

parts	<b>parameter</b>
	element <b>csta:StartDataPath</b>
source	<pre>&lt;message name="startDataPath"&gt;   &lt;part name="parameter" element="csta:StartDataPath"/&gt; &lt;/message&gt;</pre>

message **startDataPathResponse**

parts	<b>parameter</b>
	element <b>csta:StartDataPathResponse</b>
source	<pre>&lt;message name="startDataPathResponse"&gt;   &lt;part name="parameter" element="csta:StartDataPathResponse"/&gt; &lt;/message&gt;</pre>

message **stopDataPath**

parts	<b>parameter</b>
	element <b>csta:StopDataPath</b>
source	<pre>&lt;message name="stopDataPath"&gt;   &lt;part name="parameter" element="csta:StopDataPath"/&gt; &lt;/message&gt;</pre>

message **stopDataPathResponse**

parts	<b>parameter</b>
element	<b>csta:StopDataPathResponse</b>
source	<pre> &lt;message name="stopDataPathResponse"&gt;   &lt;part name="parameter" element="csta:StopDataPathResponse"/&gt; &lt;/message&gt; </pre>

message **suspendDataPath**

parts	<b>parameter</b>
element	<b>csta:SuspendDataPath</b>
source	<pre> &lt;message name="suspendDataPath"&gt;   &lt;part name="parameter" element="csta:SuspendDataPath"/&gt; &lt;/message&gt; </pre>

message **suspendDataPathResponse**

parts	<b>parameter</b>
element	<b>csta:SuspendDataPathResponse</b>
source	<pre> &lt;message name="suspendDataPathResponse"&gt;   &lt;part name="parameter" element="csta:SuspendDataPathResponse"/&gt; &lt;/message&gt; </pre>

## 18.2 Operations

operations	<b>CSTA-IO-Register</b>
input	<a href="#">tns:iORegister</a>
output	<a href="#">tns:iORegisterResponse</a>
fault	<a href="#">tns:negResponse</a>
	<b>CSTA-IO-Register-Abort</b>
input	<a href="#">tns:iORegisterAbort</a>
	<b>CSTA-IO-Register-Cancel</b>
input	<a href="#">tns:iORegisterCancel</a>
output	<a href="#">tns:iORegisterCancelResponse</a>
fault	<a href="#">tns:negResponse</a>
	<b>CSTA-Data-Path-Resumed</b>
input	<a href="#">tns:dataPathResumed</a>
output	<a href="#">tns:dataPathResumedResponse</a>
fault	<a href="#">tns:negResponse</a>

**CSTA-Data-Path-Suspended**

input [tns:dataPathSuspended](#)  
output [tns:dataPathSuspendedResponse](#)  
fault [tns:negResponse](#)

**CSTA-Fast-Data-as-input**

input [tns:fastData](#)  
output [tns:fastDataResponse](#)  
fault [tns:negResponse](#)

**CSTA-Fast-Data-as-output**

input [tns:fastData](#)  
output [tns:fastDataResponse](#)  
fault [tns:negResponse](#)

**CSTA-Resume-Data-Path**

input [tns:resumeDataPath](#)  
output [tns:resumeDataPathResponse](#)  
fault [tns:negResponse](#)

**CSTA-Send-Broadcast-Data**

input [tns:sendBroadcastData](#)  
output [tns:sendBroadcastDataResponse](#)  
fault [tns:negResponse](#)

**CSTA-Send-Data-as-input**

input [tns:sendData](#)  
output [tns:sendDataResponse](#)  
fault [tns:negResponse](#)

**CSTA-Send-Data-as-output**

input [tns:sendData](#)  
output [tns:sendDataResponse](#)  
fault [tns:negResponse](#)

**CSTA-Send-Multicast-Data**

input [tns:sendMulticastData](#)  
output [tns:sendMulticastDataResponse](#)  
fault [tns:negResponse](#)

**CSTA-Start-Data-Path-as-input**

input [tns:startDataPath](#)  
output [tns:startDataPathResponse](#)  
fault [tns:negResponse](#)



**CSTA-Start-Data-Path-as-output**

input [tns:startDataPath](#)  
 output [tns:startDataPathResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Stop-Data-Path-as-input**

input [tns:stopDataPath](#)  
 output [tns:stopDataPathResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Stop-Data-Path-as-output**

input [tns:stopDataPath](#)  
 output [tns:stopDataPathResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Suspend-Data-Path-as-input**

input [tns:suspendDataPath](#)  
 output [tns:suspendDataPathResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Suspend-Data-Path-as-output**

input [tns:suspendDataPath](#)  
 output [tns:suspendDataPathResponse](#)  
 fault [tns:negResponse](#)

source

```
<operation name="CSTA-IO-Register">
  <input message="tns:iORegister"/>
  <output message="tns:iORegisterResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-IO-Register-Abort">
  <input message="tns:iORegisterAbort"/>
</operation>
<operation name="CSTA-IO-Register-Cancel">
  <input message="tns:iORegisterCancel"/>
  <output message="tns:iORegisterCancelResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Data-Path-Resumed">
  <input message="tns:dataPathResumed"/>
  <output message="tns:dataPathResumedResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Data-Path-Suspended">
  <input message="tns:dataPathSuspended"/>
  <output message="tns:dataPathSuspendedResponse"/>
```

```

    <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Fast-Data-as-input">
    <input message="tns:fastData"/>
    <output message="tns:fastDataResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Fast-Data-as-output">
    <input message="tns:fastData"/>
    <output message="tns:fastDataResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Resume-Data-Path">
    <input message="tns:resumeDataPath"/>
    <output message="tns:resumeDataPathResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Send-Broadcast-Data">
    <input message="tns:sendBroadcastData"/>
    <output message="tns:sendBroadcastDataResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Send-Data-as-input">
    <input message="tns:sendData"/>
    <output message="tns:sendDataResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Send-Data-as-output">
    <input message="tns:sendData"/>
    <output message="tns:sendDataResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Send-Multicast-Data">
    <input message="tns:sendMulticastData"/>
    <output message="tns:sendMulticastDataResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Start-Data-Path-as-input">
    <input message="tns:startDataPath"/>
    <output message="tns:startDataPathResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Start-Data-Path-as-output">
    <input message="tns:startDataPath"/>
    <output message="tns:startDataPathResponse"/>

```

```

    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Stop-Data-Path-as-input">
    <input message="tns:stopDataPath"/>
    <output message="tns:stopDataPathResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Stop-Data-Path-as-output">
    <input message="tns:stopDataPath"/>
    <output message="tns:stopDataPathResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Suspend-Data-Path-as-input">
    <input message="tns:suspendDataPath"/>
    <output message="tns:suspendDataPathResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Suspend-Data-Path-as-output">
    <input message="tns:suspendDataPath"/>
    <output message="tns:suspendDataPathResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>

```

## 19 Data Collection Features - Messages and Operations

This Clause specifies the WSDL messages and operations for the data collection features.

### 19.1 Messages

message **dataCollected**

parts	parameter
	element <b>csta:DataCollected</b>
source	<pre> &lt;message name="dataCollected"&gt;   &lt;part name="parameter" element="csta:DataCollected"/&gt; &lt;/message&gt; </pre>

message **dataCollectedResponse**

parts	parameter
	element <b>csta:DataCollectedResponse</b>
source	<pre> &lt;message name="dataCollectedResponse"&gt;   &lt;part name="parameter" element="csta:DataCollectedResponse"/&gt; &lt;/message&gt; </pre>

message **dataCollectionResumed**

parts	<b>parameter</b>
	element <b>csta:DataCollectionResumed</b>
source	<pre>&lt;message name="dataCollectionResumed"&gt;   &lt;part name="parameter" element="csta:DataCollectionResumed"/&gt; &lt;/message&gt;</pre>

message **dataCollectionResumedResponse**

parts	<b>parameter</b>
	element <b>csta:DataCollectionResumedResponse</b>
source	<pre>&lt;message name="dataCollectionResumedResponse"&gt;   &lt;part name="parameter" element="csta:DataCollectionResumedResponse"/&gt; &lt;/message&gt;</pre>

message **dataCollectionSuspended**

parts	<b>parameter</b>
	element <b>csta:DataCollectionSuspended</b>
source	<pre>&lt;message name="dataCollectionSuspended"&gt;   &lt;part name="parameter" element="csta:DataCollectionSuspended"/&gt; &lt;/message&gt;</pre>

message **dataCollectionSuspendedResponse**

parts	<b>parameter</b>
	element <b>csta:DataCollectionSuspendedResponse</b>
source	<pre>&lt;message name="dataCollectionSuspendedResponse"&gt;   &lt;part name="parameter" element="csta:DataCollectionSuspendedResponse"/&gt; &lt;/message&gt;</pre>

message **resumeDataCollection**

parts	<b>parameter</b>
	element <b>csta:ResumeDataCollection</b>
source	<pre>&lt;message name="resumeDataCollection"&gt;   &lt;part name="parameter" element="csta:ResumeDataCollection"/&gt; &lt;/message&gt;</pre>

message **resumeDataCollectionResponse**

parts	<b>parameter</b>
	element <b>csta:ResumeDataCollectionResponse</b>
source	<pre>&lt;message name="resumeDataCollectionResponse"&gt;   &lt;part name="parameter" element="csta:ResumeDataCollectionResponse"/&gt; &lt;/message&gt;</pre>

message **startDataCollection**

parts	<b>parameter</b>
	element <b>csta:StartDataCollection</b>
source	<pre>&lt;message name="startDataCollection"&gt;   &lt;part name="parameter" element="csta:StartDataCollection"/&gt; &lt;/message&gt;</pre>

message **startDataCollectionResponse**

parts	<b>parameter</b>
	element <b>csta:StartDataCollectionResponse</b>
source	<pre>&lt;message name="startDataCollectionResponse"&gt;   &lt;part name="parameter" element="csta:StartDataCollectionResponse"/&gt; &lt;/message&gt;</pre>

message **stopDataCollection**

parts	<b>parameter</b>
	element <b>csta:StopDataCollection</b>
source	<pre>&lt;message name="stopDataCollection"&gt;   &lt;part name="parameter" element="csta:StopDataCollection"/&gt; &lt;/message&gt;</pre>

message **stopDataCollectionResponse**

parts	<b>parameter</b>
	element <b>csta:StopDataCollectionResponse</b>
source	<pre>&lt;message name="stopDataCollectionResponse"&gt;   &lt;part name="parameter" element="csta:StopDataCollectionResponse"/&gt; &lt;/message&gt;</pre>

message **suspendDataCollection**

parts	<b>parameter</b>
	element <b>csta:SuspendDataCollection</b>
source	<pre>&lt;message name="suspendDataCollection"&gt;   &lt;part name="parameter" element="csta:SuspendDataCollection"/&gt; &lt;/message&gt;</pre>

message **suspendDataCollectionResponse**

parts	<b>parameter</b>
	element <b>csta:SuspendDataCollectionResponse</b>
source	<pre>&lt;message name="suspendDataCollectionResponse"&gt;   &lt;part name="parameter" element="csta:SuspendDataCollectionResponse"/&gt; &lt;/message&gt;</pre>

## 19.2 Operations

operations	<b>CSTA-Data-Collected</b>
	input <a href="#">tns:dataCollected</a>
	output <a href="#">tns:dataCollectedResponse</a>
	fault <a href="#">tns:negResponse</a>
	<b>CSTA-Data-Collection-Resumed</b>
	input <a href="#">tns:dataCollectionResumed</a>
	output <a href="#">tns:dataCollectionResumedResponse</a>
	fault <a href="#">tns:negResponse</a>
	<b>CSTA-Data-Collection-Suspended</b>
	input <a href="#">tns:dataCollectionSuspended</a>
	output <a href="#">tns:dataCollectionSuspendedResponse</a>
	fault <a href="#">tns:negResponse</a>
	<b>CSTA-Resume-Data-Collection</b>
	input <a href="#">tns:resumeDataCollection</a>
	output <a href="#">tns:resumeDataCollectionResponse</a>
	fault <a href="#">tns:negResponse</a>
	<b>CSTA-Start-Data-Collection</b>
	input <a href="#">tns:startDataCollection</a>
	output <a href="#">tns:startDataCollectionResponse</a>
	fault <a href="#">tns:negResponse</a>

**CSTA-Stop-Data-Collection-as-input**

input [tns:stopDataCollection](#)  
 output [tns:stopDataCollectionResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Stop-Data-Collection-as-output**

input [tns:stopDataCollection](#)  
 output [tns:stopDataCollectionResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Suspend-Data-Collection-as-input**

input [tns:suspendDataCollection](#)  
 output [tns:suspendDataCollectionResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Suspend-Data-Collection-as-output**

input [tns:suspendDataCollection](#)  
 output [tns:suspendDataCollectionResponse](#)  
 fault [tns:negResponse](#)

source

```
<operation name="CSTA-Data-Collected">
  <input message="tns:dataCollected"/>
  <output message="tns:dataCollectedResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Data-Collection-Resumed">
  <input message="tns:dataCollectionResumed"/>
  <output message="tns:dataCollectionResumedResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Data-Collection-Suspended">
  <input message="tns:dataCollectionSuspended"/>
  <output message="tns:dataCollectionSuspendedResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Resume-Data-Collection">
  <input message="tns:resumeDataCollection"/>
  <output message="tns:resumeDataCollectionResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Start-Data-Collection">
  <input message="tns:startDataCollection"/>
  <output message="tns:startDataCollectionResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Stop-Data-Collection-as-input">
```

```

<input message="tns:stopDataCollection"/>
<output message="tns:stopDataCollectionResponse"/>
<fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Stop-Data-Collection-as-output">
  <input message="tns:stopDataCollection"/>
  <output message="tns:stopDataCollectionResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Suspend-Data-Collection-as-input">
  <input message="tns:suspendDataCollection"/>
  <output message="tns:suspendDataCollectionResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Suspend-Data-Collection-as-output">
  <input message="tns:suspendDataCollection"/>
  <output message="tns:suspendDataCollectionResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

```

## 20 Voice Features - Messages and Operations

This Clause specifies the WSDL messages and operations for the Voice Services.

### 20.1 Messages

#### message activate

parts	<b>parameter</b>
	element <b>csta:Activate</b>
source	<pre> &lt;message name="activate"&gt;   &lt;part name="parameter" element="csta:Activate"/&gt; &lt;/message&gt; </pre>

#### message activateResponse

parts	<b>parameter</b>
	element <b>csta:ActivateResponse</b>
source	<pre> &lt;message name="activateResponse"&gt;   &lt;part name="parameter" element="csta:ActivateResponse"/&gt; &lt;/message&gt; </pre>



**message clear**

parts	<b>parameter</b>
	element <b>csta:Clear</b>
source	<message name="clear">
	<part name="parameter" element="csta:Clear"/>
	</message>

**message clearResponse**

parts	<b>parameter</b>
	element <b>csta:ClearResponse</b>
source	<message name="clearResponse">
	<part name="parameter" element="csta:ClearResponse"/>
	</message>

**message concatenateMessage**

parts	<b>parameter</b>
	element <b>csta:ConcatenateMessage</b>
source	<message name="concatenateMessage">
	<part name="parameter" element="csta:ConcatenateMessage"/>
	</message>

**message concatenateMessageResponse**

parts	<b>parameter</b>
	element <b>csta:ConcatenateMessageResponse</b>
source	<message name="concatenateMessageResponse">
	<part name="parameter" element="csta:ConcatenateMessageResponse"/>
	</message>

**message deactivate**

parts	<b>parameter</b>
	element <b>csta:Deactivate</b>
source	<message name="deactivate">
	<part name="parameter" element="csta:Deactivate"/>
	</message>

message **deactivateResponse**

parts	<b>parameter</b>
	element <b>csta:DeactivateResponse</b>
source	<message name="deactivateResponse">
	<part name="parameter" element="csta:DeactivateResponse"/>
	</message>

message **deleteMessage**

parts	<b>parameter</b>
	element <b>csta&gt;DeleteMessage</b>
source	<message name="deleteMessage">
	<part name="parameter" element="csta>DeleteMessage"/>
	</message>

message **deleteMessageResponse**

parts	<b>parameter</b>
	element <b>csta&gt;DeleteMessageResponse</b>
source	<message name="deleteMessageResponse">
	<part name="parameter" element="csta>DeleteMessageResponse"/>
	</message>

message **playMessage**

parts	<b>parameter</b>
	element <b>csta:PlayMessage</b>
source	<message name="playMessage">
	<part name="parameter" element="csta:PlayMessage"/>
	</message>

message **playMessageResponse**

parts	<b>parameter</b>
	element <b>csta:PlayMessageResponse</b>
source	<message name="playMessageResponse">
	<part name="parameter" element="csta:PlayMessageResponse"/>
	</message>

**message queryVoiceAttribute**

parts	<b>parameter</b>
	element <b>csta:QueryVoiceAttribute</b>
source	<pre>&lt;message name="queryVoiceAttribute"&gt;   &lt;part name="parameter" element="csta:QueryVoiceAttribute"/&gt; &lt;/message&gt;</pre>

**message queryVoiceAttributeResponse**

parts	<b>parameter</b>
	element <b>csta:QueryVoiceAttributeResponse</b>
source	<pre>&lt;message name="queryVoiceAttributeResponse"&gt;   &lt;part name="parameter" element="csta:QueryVoiceAttributeResponse"/&gt; &lt;/message&gt;</pre>

**message queueService**

parts	<b>parameter</b>
	element <b>csta:QueueService</b>
source	<pre>&lt;message name="queueService"&gt;   &lt;part name="parameter" element="csta:QueueService"/&gt; &lt;/message&gt;</pre>

**message queueServiceResponse**

parts	<b>parameter</b>
	element <b>csta:QueueServiceResponse</b>
source	<pre>&lt;message name="queueServiceResponse"&gt;   &lt;part name="parameter" element="csta:QueueServiceResponse"/&gt; &lt;/message&gt;</pre>

**message recordMessage**

parts	<b>parameter</b>
	element <b>csta:RecordMessage</b>
source	<pre>&lt;message name="recordMessage"&gt;   &lt;part name="parameter" element="csta:RecordMessage"/&gt; &lt;/message&gt;</pre>

message **recordMessageResponse**

parts	<b>parameter</b>
	element <b>csta:RecordMessageResponse</b>
source	<code>&lt;message name="recordMessageResponse"&gt;</code>
	<code>&lt;part name="parameter" element="csta:RecordMessageResponse"/&gt;</code>
	<code>&lt;/message&gt;</code>

message **reposition**

parts	<b>parameter</b>
	element <b>csta:Reposition</b>
source	<code>&lt;message name="reposition"&gt;</code>
	<code>&lt;part name="parameter" element="csta:Reposition"/&gt;</code>
	<code>&lt;/message&gt;</code>

message **repositionResponse**

parts	<b>parameter</b>
	element <b>csta:RepositionResponse</b>
source	<code>&lt;message name="repositionResponse"&gt;</code>
	<code>&lt;part name="parameter" element="csta:RepositionResponse"/&gt;</code>
	<code>&lt;/message&gt;</code>

message **resume**

parts	<b>parameter</b>
	element <b>csta:Resume</b>
source	<code>&lt;message name="resume"&gt;</code>
	<code>&lt;part name="parameter" element="csta:Resume"/&gt;</code>
	<code>&lt;/message&gt;</code>

message **resumeResponse**

parts	<b>parameter</b>
	element <b>csta:ResumeResponse</b>
source	<code>&lt;message name="resumeResponse"&gt;</code>
	<code>&lt;part name="parameter" element="csta:ResumeResponse"/&gt;</code>
	<code>&lt;/message&gt;</code>

message **review**

parts	<b>parameter</b>
	element <b>csta:Review</b>
source	<pre>&lt;message name="review"&gt;   &lt;part name="parameter" element="csta:Review"/&gt; &lt;/message&gt;</pre>

message **reviewResponse**

parts	<b>parameter</b>
	element <b>csta:ReviewResponse</b>
source	<pre>&lt;message name="reviewResponse"&gt;   &lt;part name="parameter" element="csta:ReviewResponse"/&gt; &lt;/message&gt;</pre>

message **setVoiceAttribute**

parts	<b>parameter</b>
	element <b>csta:SetVoiceAttribute</b>
source	<pre>&lt;message name="setVoiceAttribute"&gt;   &lt;part name="parameter" element="csta:SetVoiceAttribute"/&gt; &lt;/message&gt;</pre>

message **setVoiceAttributeResponse**

parts	<b>parameter</b>
	element <b>csta:SetVoiceAttributeResponse</b>
source	<pre>&lt;message name="setVoiceAttributeResponse"&gt;   &lt;part name="parameter" element="csta:SetVoiceAttributeResponse"/&gt; &lt;/message&gt;</pre>

message **start**

parts	<b>parameter</b>
	element <b>csta:Start</b>
source	<pre>&lt;message name="start"&gt;   &lt;part name="parameter" element="csta:Start"/&gt; &lt;/message&gt;</pre>

message **startResponse**

parts	<b>parameter</b>
	element <b>csta:StartResponse</b>
source	<message name="startResponse">
	<part name="parameter" element="csta:StartResponse"/>
	</message>

message **stop**

parts	<b>parameter</b>
	element <b>csta:Stop</b>
source	<message name="stop">
	<part name="parameter" element="csta:Stop"/>
	</message>

message **stopResponse**

parts	<b>parameter</b>
	element <b>csta:StopResponse</b>
source	<message name="stopResponse">
	<part name="parameter" element="csta:StopResponse"/>
	</message>

message **suspend**

parts	<b>parameter</b>
	element <b>csta:Suspend</b>
source	<message name="suspend">
	<part name="parameter" element="csta:Suspend"/>
	</message>

message **suspendResponse**

parts	<b>parameter</b>
	element <b>csta:SuspendResponse</b>
source	<message name="suspendResponse">
	<part name="parameter" element="csta:SuspendResponse"/>
	</message>

message **synthesizeMessage**

parts	<b>parameter</b>
	element <b>csta:SynthesizeMessage</b>
source	<pre>&lt;message name="synthesizeMessage"&gt;   &lt;part name="parameter" element="csta:SynthesizeMessage"/&gt; &lt;/message&gt;</pre>

message **synthesizeMessageResponse**

parts	<b>parameter</b>
	element <b>csta:SynthesizeMessageResponse</b>
source	<pre>&lt;message name="synthesizeMessageResponse"&gt;   &lt;part name="parameter" element="csta:SynthesizeMessageResponse"/&gt; &lt;/message&gt;</pre>

message **bookmarkReachedEvent**

parts	<b>parameter</b>
	element <b>csta:BookmarkReachedEvent</b>
source	<pre>&lt;message name="bookmarkReachedEvent"&gt;   &lt;part name="parameter" element="csta:BookmarkReachedEvent"/&gt; &lt;/message&gt;</pre>

message **completedEvent**

parts	<b>parameter</b>
	element <b>csta:CompletedEvent</b>
source	<pre>&lt;message name="completedEvent"&gt;   &lt;part name="parameter" element="csta:CompletedEvent"/&gt; &lt;/message&gt;</pre>

message **dtmfDetectedEvent**

parts	<b>parameter</b>
	element <b>csta:DtmfDetectedEvent</b>
source	<pre>&lt;message name="dtmfDetectedEvent"&gt;   &lt;part name="parameter" element="csta:DtmfDetectedEvent"/&gt; &lt;/message&gt;</pre>

message **emptiedEvent**

parts	<b>parameter</b>
	element <b>csta:EmptiedEvent</b>
source	<message name="emptiedEvent">
	<part name="parameter" element="csta:EmptiedEvent"/>
	</message>

message **interruptionDetectedEvent**

parts	<b>parameter</b>
	element <b>csta:InterruptionDetectedEvent</b>
source	<message name="interruptionDetectedEvent">
	<part name="parameter" element="csta:InterruptionDetectedEvent"/>
	</message>

message **notRecognizedEvent**

parts	<b>parameter</b>
	element <b>csta:NotRecognizedEvent</b>
source	<message name="notRecognizedEvent">
	<part name="parameter" element="csta:NotRecognizedEvent"/>
	</message>

message **playEvent**

parts	<b>parameter</b>
	element <b>csta:PlayEvent</b>
source	<message name="playEvent">
	<part name="parameter" element="csta:PlayEvent"/>
	</message>

message **recognizedEvent**

parts	<b>parameter</b>
	element <b>csta:RecognizedEvent</b>
source	<message name="recognizedEvent">
	<part name="parameter" element="csta:RecognizedEvent"/>
	</message>



message **recordEvent**

parts	<b>parameter</b>
	element <b>csta:RecordEvent</b>
source	<pre> &lt;message name="recordEvent"&gt;   &lt;part name="parameter" element="csta:RecordEvent"/&gt; &lt;/message&gt; </pre>

message **reviewEvent**

parts	<b>parameter</b>
	element <b>csta:ReviewEvent</b>
source	<pre> &lt;message name="reviewEvent"&gt;   &lt;part name="parameter" element="csta:ReviewEvent"/&gt; &lt;/message&gt; </pre>

message **silenceTimeoutExpiredEvent**

parts	<b>parameter</b>
	element <b>csta:SilenceTimeoutExpiredEvent</b>
source	<pre> &lt;message name="silenceTimeoutExpiredEvent"&gt;   &lt;part name="parameter" element="csta:SilenceTimeoutExpiredEvent"/&gt; &lt;/message&gt; </pre>

message **speechDetectedEvent**

parts	<b>parameter</b>
	element <b>csta:SpeechDetectedEvent</b>
source	<pre> &lt;message name="speechDetectedEvent"&gt;   &lt;part name="parameter" element="csta:SpeechDetectedEvent"/&gt; &lt;/message&gt; </pre>

message **startedEvent**

parts	<b>parameter</b>
	element <b>csta:StartedEvent</b>
source	<pre> &lt;message name="startedEvent"&gt;   &lt;part name="parameter" element="csta:StartedEvent"/&gt; &lt;/message&gt; </pre>

message **stopEvent**

parts	<b>parameter</b>
	element <b>csta:StopEvent</b>
source	<message name="stopEvent">
	<part name="parameter" element="csta:StopEvent"/>
	</message>

message **suspendPlayEvent**

parts	<b>parameter</b>
	element <b>csta:SuspendPlayEvent</b>
source	<message name="suspendPlayEvent">
	<part name="parameter" element="csta:SuspendPlayEvent"/>
	</message>

message **suspendRecordEvent**

parts	<b>parameter</b>
	element <b>csta:SuspendRecordEvent</b>
source	<message name="suspendRecordEvent">
	<part name="parameter" element="csta:SuspendRecordEvent"/>
	</message>

message **voiceAttributeChangedEvent**

parts	<b>parameter</b>
	element <b>csta:VoiceAttributeChangedEvent</b>
source	<message name="voiceAttributeChangedEvent">
	<part name="parameter" element="csta:VoiceAttributeChangedEvent"/>
	</message>

message **voiceErrorOccurredEvent**

parts	<b>parameter</b>
	element <b>csta:VoiceErrorOccurredEvent</b>
source	<message name="voiceErrorOccurredEvent">
	<part name="parameter" element="csta:VoiceErrorOccurredEvent"/>
	</message>

## 20.2 Operations

operations

### CSTA-Activate

input [tns:activate](#)  
 output [tns:activateResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Clear

input [tns:clear](#)  
 output [tns:clearResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Concatenate-Message

input [tns:concatenateMessage](#)  
 output [tns:concatenateMessageResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Deactivate

input [tns:deactivate](#)  
 output [tns:deactivateResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Delete-Message

input [tns:deleteMessage](#)  
 output [tns:deleteMessageResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Play-Message

input [tns:playMessage](#)  
 output [tns:playMessageResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Query-Voice-Attribute

input [tns:queryVoiceAttribute](#)  
 output [tns:queryVoiceAttributeResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Queue-Service

input [tns:queueService](#)  
 output [tns:queueServiceResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Record-Message

input [tns:recordMessage](#)  
 output [tns:recordMessageResponse](#)  
 fault [tns:negResponse](#)

#### CSTA-Reposition

input [tns:reposition](#)  
output [tns:repositionResponse](#)  
fault [tns:negResponse](#)

#### CSTA-Resume

input [tns:resume](#)  
output [tns:resumeResponse](#)  
fault [tns:negResponse](#)

#### CSTA-Review

input [tns:review](#)  
output [tns:reviewResponse](#)  
fault [tns:negResponse](#)

#### CSTA-Set-Voice-Attribute

input [tns:setVoiceAttribute](#)  
output [tns:setVoiceAttributeResponse](#)  
fault [tns:negResponse](#)

#### CSTA-Start

input [tns:start](#)  
output [tns:startResponse](#)  
fault [tns:negResponse](#)

#### CSTA-Stop

input [tns:stop](#)  
output [tns:stopResponse](#)  
fault [tns:negResponse](#)

#### CSTA-Suspend

input [tns:suspend](#)  
output [tns:suspendResponse](#)  
fault [tns:negResponse](#)

#### CSTA-Synthesize-Message

input [tns:synthesizeMessage](#)  
output [tns:synthesizeMessageResponse](#)  
fault [tns:negResponse](#)

#### CSTA-Bookmark-Reached-Event

inp [tns:bookmarkReachedEvent](#)

#### CSTA-Completed-Event

input [tns:completedEvent](#)

**CSTA-Dtmf-Detected-Event**input [tns:dtmfDetectedEvent](#)**CSTA-Emptied-Event**input [tns:emptiedEvent](#)**CSTA-Interruption-Detected-Event**input [tns:interruptionDetectedEvent](#)**CSTA-Not-Recognized-Event**input [tns:notRecognizedEvent](#)**CSTA-Play-Event**input [tns:playEvent](#)**CSTA-Recognized-Event**input [tns:recognizedEvent](#)**CSTA-Record-Event**input [tns:recordEvent](#)**CSTA-Review-Event**input [tns:reviewEvent](#)**CSTA-Silence-Timeout-Expired-Event**input [tns:silenceTimeoutExpiredEvent](#)**CSTA-Speech-Detected-Event**input [tns:speechDetectedEvent](#)**CSTA-Started-Event**input [tns:startedEvent](#)**CSTA-Stop-Event**input [tns:stopEvent](#)**CSTA-Suspend-Play-Event**input [tns:suspendPlayEvent](#)**CSTA-Suspend-Record-Event**input [tns:suspendRecordEvent](#)

**CSTA-Voice-Attribute-Changed-Event**input [tns:voiceAttributeChangedEvent](#)**CSTA-Voice-Error-Occurred-Event**input [tns:voiceErrorOccurredEvent](#)

```

source
<operation name="CSTA-Activate">
  <input message="tns:activate"/>
  <output message="tns:activateResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Clear">
  <input message="tns:clear"/>
  <output message="tns:clearResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Concatenate-Message">
  <input message="tns:concatenateMessage"/>
  <output message="tns:concatenateMessageResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Deactivate">
  <input message="tns:deactivate"/>
  <output message="tns:deactivateResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Delete-Message">
  <input message="tns:deleteMessage"/>
  <output message="tns:deleteMessageResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Play-Message">
  <input message="tns:playMessage"/>
  <output message="tns:playMessageResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Query-Voice-Attribute">
  <input message="tns:queryVoiceAttribute"/>
  <output message="tns:queryVoiceAttributeResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Queue-Service">
  <input message="tns:queueService"/>
  <output message="tns:queueServiceResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

```

```

<operation name="CSTA-Record-Message">
  <input message="tns:recordMessage"/>
  <output message="tns:recordMessageResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Reposition">
  <input message="tns:reposition"/>
  <output message="tns:repositionResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Resume">
  <input message="tns:resume"/>
  <output message="tns:resumeResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Review">
  <input message="tns:review"/>
  <output message="tns:reviewResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Set-Voice-Attribute">
  <input message="tns:setVoiceAttribute"/>
  <output message="tns:setVoiceAttributeResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Start">
  <input message="tns:start"/>
  <output message="tns:startResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Stop">
  <input message="tns:stop"/>
  <output message="tns:stopResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Suspend">
  <input message="tns:suspend"/>
  <output message="tns:suspendResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Synthesize-Message">
  <input message="tns:synthesizeMessage"/>
  <output message="tns:synthesizeMessageResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

```

```

<operation name="CSTA-Bookmark-Reached-Event">
  <input message="tns:bookmarkReachedEvent"/>
</operation>
<operation name="CSTA-Completed-Event">
  <input message="tns:completedEvent"/>
</operation>
<operation name="CSTA-Dtmf-Detected-Event">
  <input message="tns:dtmfDetectedEvent"/>
</operation>
<operation name="CSTA-Emptied-Event">
  <input message="tns:emptiedEvent"/>
</operation>
<operation name="CSTA-Interruption-Detected-Event">
  <input message="tns:interruptionDetectedEvent"/>
</operation>
<operation name="CSTA-Not-Recognized-Event">
  <input message="tns:notRecognizedEvent"/>
</operation>
<operation name="CSTA-Play-Event">
  <input message="tns:playEvent"/>
</operation>
<operation name="CSTA-Recognized-Event">
  <input message="tns:recognizedEvent"/>
</operation>
<operation name="CSTA-Record-Event">
  <input message="tns:recordEvent"/>
</operation>
<operation name="CSTA-Review-Event">
  <input message="tns:reviewEvent"/>
</operation>
<operation name="CSTA-Silence-Timeout-Expired-Event">
  <input message="tns:silenceTimeoutExpiredEvent"/>
</operation>
<operation name="CSTA-Speech-Detected-Event">
  <input message="tns:speechDetectedEvent"/>
</operation>
<operation name="CSTA-Started-Event">
  <input message="tns:startedEvent"/>
</operation>
<operation name="CSTA-Stop-Event">
  <input message="tns:stopEvent"/>
</operation>
<operation name="CSTA-Suspend-Play-Event">
  <input message="tns:suspendPlayEvent"/>
</operation>

```



```

<operation name="CSTA-Suspend-Record-Event">
  <input message="tns:suspendRecordEvent"/>
</operation>
<operation name="CSTA-Voice-Attribute-Changed-Event">
  <input message="tns:voiceAttributeChangedEvent"/>
</operation>
<operation name="CSTA-Voice-Error-Occurred-Event">
  <input message="tns:voiceErrorOccurredEvent"/>
</operation>

```

## 21 Call Detail Record Features - Messages and Operations

This Clause specifies the WSDL messages and operations for the Call Detail Record features.

### 21.1 Messages

message **cDRNotification**

parts	parameter
	element <b>csta:CDRNotification</b>
source	<pre> &lt;message name="cDRNotification"&gt;   &lt;part name="parameter" element="csta:CDRNotification"/&gt; &lt;/message&gt; </pre>

message **cDRNotificationResponse**

parts	parameter
	element <b>csta:CDRNotificationResponse</b>
source	<pre> &lt;message name="cDRNotificationResponse"&gt;   &lt;part name="parameter" element="csta:CDRNotificationResponse"/&gt; &lt;/message&gt; </pre>

message **cDRReport**

parts	parameter
	element <b>csta:CDRReport</b>
source	<pre> &lt;message name="cDRReport"&gt;   &lt;part name="parameter" element="csta:CDRReport"/&gt; &lt;/message&gt; </pre>

message **cDRReportResponse**

parts	parameter
	element <b>csta:CDRReportResponse</b>
source	<pre> &lt;message name="cDRReportResponse"&gt; </pre>

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

#### message **sendStoredCDR**

parts	<b>parameter</b>
	element <b>csta:SendStoredCDR</b>
source	<message name="sendStoredCDR">
	<part name="parameter" element="csta:SendStoredCDR"/>
	</message>

#### message **sendStoredCDRResponse**

parts	<b>parameter</b>
	element <b>csta:SendStoredCDRResponse</b>
source	<message name="sendStoredCDRResponse">
	<part name="parameter" element="csta:SendStoredCDRResponse"/>
	</message>

#### message **startCDRTransmission**

parts	<b>parameter</b>
	element <b>csta:StartCDRTransmission</b>
source	<message name="startCDRTransmission">
	<part name="parameter" element="csta:StartCDRTransmission"/>
	</message>

#### message **startCDRTransmissionResponse**

parts	<b>parameter</b>
	element <b>csta:StartCDRTransmissionResponse</b>
source	<message name="startCDRTransmissionResponse">
	<part name="parameter" element="csta:StartCDRTransmissionResponse"/>
	</message>

#### message **stopCDRTransmission**

parts	<b>parameter</b>
	element <b>csta:StopCDRTransmission</b>
source	<message name="stopCDRTransmission">
	<part name="parameter" element="csta:StopCDRTransmission"/>
	</message>

message **stopCDRTransmissionResponse**

parts	<b>parameter</b>
element	<b>csta:StopCDRTransmissionResponse</b>
source	<pre> &lt;message name="stopCDRTransmissionResponse"&gt;   &lt;part name="parameter" element="csta:StopCDRTransmissionResponse"/&gt; &lt;/message&gt; </pre>

## 21.2 Operations

operations	<b>CSTA-CDR-Notification</b>
input	<a href="#">tns:cDRNotification</a>
output	<a href="#">tns:cDRNotificationResponse</a>
fault	<a href="#">tns:negResponse</a>
	<b>CSTA-CDR-Report</b>
input	<a href="#">tns:cDRReport</a>
output	<a href="#">tns:cDRReportResponse</a>
fault	<a href="#">tns:negResponse</a>
	<b>CSTA-Send-Stored-CDR</b>
input	<a href="#">tns:sendStoredCDR</a>
output	<a href="#">tns:sendStoredCDRResponse</a>
fault	<a href="#">tns:negResponse</a>
	<b>CSTA-Start-CDR-Transmission</b>
input	<a href="#">tns:startCDRTransmission</a>
output	<a href="#">tns:startCDRTransmissionResponse</a>
fault	<a href="#">tns:negResponse</a>
	<b>CSTA-Stop-CDR-Transmission-as-input</b>
input	<a href="#">tns:stopCDRTransmission</a>
output	<a href="#">tns:stopCDRTransmissionResponse</a>
fault	<a href="#">tns:negResponse</a>
	<b>CSTA-Stop-CDR-Transmission-as-output</b>
input	<a href="#">tns:stopCDRTransmission</a>
output	<a href="#">tns:stopCDRTransmissionResponse</a>
fault	<a href="#">tns:negResponse</a>
source	<pre> &lt;operation name="CSTA-CDR-Notification"&gt;   &lt;input message="tns:cDRNotification"/&gt;   &lt;output message="tns:cDRNotificationResponse"/&gt; </pre>

```

    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-CDR-Report">
    <input message="tns:cDRReport"/>
    <output message="tns:cDRReportResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Send-Stored-CDR">
    <input message="tns:sendStoredCDR"/>
    <output message="tns:sendStoredCDRResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Start-CDR-Transmission">
    <input message="tns:startCDRTransmission"/>
    <output message="tns:startCDRTransmissionResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Stop-CDR-Transmission-as-input">
    <input message="tns:stopCDRTransmission"/>
    <output message="tns:stopCDRTransmissionResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Stop-CDR-Transmission-as-output">
    <input message="tns:stopCDRTransmission"/>
    <output message="tns:stopCDRTransmissionResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>

```

## 22 Location Services - Messages and Operations

This Clause specifies the WSDL messages and operations for the Location Services.

### 22.1 Messages

message **getLocationInformation**

parts	<b>parameter</b>
	element <b>csta:GetLocationInformation</b>
source	<pre> &lt;message name="getLocationInformation"&gt;   &lt;part name="parameter" element="csta:GetLocationInformation"/&gt; &lt;/message&gt; </pre>

message **getLocationInformationResponse**

parts	<b>parameter</b>
	element <b>csta:getLocationInformationResponse</b>
source	<pre>&lt;message name="getLocationInformationResponse"&gt;   &lt;part name="parameter" element="csta:getLocationInformationResponse"/&gt; &lt;/message&gt;</pre>

message **setLocationInformation**

parts	<b>parameter</b>
	element <b>csta:SetLocationInformation</b>
source	<pre>&lt;message name="setLocationInformation"&gt;   &lt;part name="parameter" element="csta:SetLocationInformation"/&gt; &lt;/message&gt;</pre>

message **setLocationInformationResponse**

parts	<b>parameter</b>
	element <b>csta:SetLocationInformationResponse</b>
source	<pre>&lt;message name="setLocationInformationResponse"&gt;   &lt;part name="parameter" element="csta:SetLocationInformationResponse"/&gt; &lt;/message&gt;</pre>

message **locationTrackingSessionResumed**

parts	<b>parameter</b>
	element <b>csta:LocationTrackingSessionResumed</b>
source	<pre>&lt;message name="locationTrackingSessionResumed"&gt;   &lt;part name="parameter" element="csta:LocationTrackingSessionResumed"/&gt; &lt;/message&gt;</pre>

message **locationTrackingSessionResumedResponse**

parts	<b>parameter</b>
	element <b>csta:LocationTrackingSessionResumedResponse</b>
source	<pre>&lt;message name="locationTrackingSessionResumedResponse"&gt;   &lt;part name="parameter" element="csta:LocationTrackingSessionResumedResponse"/&gt; &lt;/message&gt;</pre>

message **locationTrackingSessionSuspended**

parts	<b>parameter</b>
	element <b>csta:LocationTrackingSessionSuspended</b>
source	<code>&lt;message name="locationTrackingSessionSuspended"&gt;</code>
	<code>&lt;part name="parameter" element="csta:LocationTrackingSessionSuspended"/&gt;</code>
	<code>&lt;/message&gt;</code>

message **locationTrackingSessionSuspendedResponse**

parts	<b>parameter</b>
	element <b>csta:LocationTrackingSessionSuspendedResponse</b>
source	<code>&lt;message name="locationTrackingSessionSuspendedResponse"&gt;</code>
	<code>&lt;part name="parameter" element="csta:LocationTrackingSessionSuspendedResponse"/&gt;</code>
	<code>&lt;/message&gt;</code>

message **resumeLocationTrackingSession**

parts	<b>parameter</b>
	element <b>csta:ResumeLocationTrackingSession</b>
source	<code>&lt;message name="resumeLocationTrackingSession"&gt;</code>
	<code>&lt;part name="parameter" element="csta:ResumeLocationTrackingSession"/&gt;</code>
	<code>&lt;/message&gt;</code>

message **resumeLocationTrackingSessionResponse**

parts	<b>parameter</b>
	element <b>csta:ResumeLocationTrackingSessionResponse</b>
source	<code>&lt;message name="resumeLocationTrackingSessionResponse"&gt;</code>
	<code>&lt;part name="parameter" element="csta:ResumeLocationTrackingSessionResponse"/&gt;</code>
	<code>&lt;/message&gt;</code>

message **locationInformationReport**

parts	<b>parameter</b>
	element <b>csta:LocationInformationReport</b>
source	<code>&lt;message name="locationInformationReport"&gt;</code>
	<code>&lt;part name="parameter" element="csta:LocationInformationReport"/&gt;</code>
	<code>&lt;/message&gt;</code>

message **locationInformationReportResponse**

parts	<b>parameter</b>
	element <b>csta:LocationInformationReportResponse</b>
source	<code>&lt;message name="locationInformationReportResponse"&gt;</code>
	<code>&lt;part name="parameter" element="csta:LocationInformationReportResponse"/&gt;</code>
	<code>&lt;/message&gt;</code>

message **startLocationTrackingSession**

parts	<b>parameter</b>
	element <b>csta:StartLocationTrackingSession</b>
source	<code>&lt;message name="startLocationTrackingSession"&gt;</code>
	<code>&lt;part name="parameter" element="csta:StartLocationTrackingSession"/&gt;</code>
	<code>&lt;/message&gt;</code>

message **startLocationTrackingSessionResponse**

parts	<b>parameter</b>
	element <b>csta:StartLocationTrackingSessionResponse</b>
source	<code>&lt;message name="startLocationTrackingSessionResponse"&gt;</code>
	<code>&lt;part name="parameter" element="csta:StartLocationTrackingSessionResponse"/&gt;</code>
	<code>&lt;/message&gt;</code>

message **stopLocationTrackingSession**

parts	<b>parameter</b>
	element <b>csta:StopLocationTrackingSession</b>
source	<code>&lt;message name="stopLocationTrackingSession"&gt;</code>
	<code>&lt;part name="parameter" element="csta:StopLocationTrackingSession"/&gt;</code>
	<code>&lt;/message&gt;</code>

message **stopLocationTrackingSessionResponse**

parts	<b>parameter</b>
	element <b>csta:StopLocationTrackingSessionResponse</b>
source	<code>&lt;message name="stopLocationTrackingSessionResponse"&gt;</code>
	<code>&lt;part name="parameter" element="csta:StopLocationTrackingSessionResponse"/&gt;</code>
	<code>&lt;/message&gt;</code>

message **suspendLocationTrackingSession**

parts	<b>parameter</b>
	element <b>csta:SuspendLocationTrackingSession</b>
source	<message name="suspendLocationTrackingSession">
	<part name="parameter" element="csta:SuspendLocationTrackingSession"/>
	</message>

message **suspendLocationTrackingSessionResponse**

parts	<b>parameter</b>
	element <b>csta:SuspendLocationTrackingSessionResponse</b>
source	<message name="suspendLocationTrackingSessionResponse">
	<part name="parameter" element="csta:SuspendLocationTrackingSessionResponse"/>
	</message>

message **getLocationTrackingCapabilities**

parts	<b>parameter</b>
	element <b>csta:GetLocationTrackingCapabilities</b>
source	<message name="getLocationTrackingCapabilities">
	<part name="parameter" element="csta:GetLocationTrackingCapabilities"/>
	</message>

message **getLocationTrackingCapabilitiesResponse**

parts	<b>parameter</b>
	element <b>csta:GetLocationTrackingCapabilitiesResponse</b>
source	<message name="getLocationTrackingCapabilitiesResponse">
	<part name="parameter" element="csta:GetLocationTrackingCapabilitiesResponse"/>
	</message>

message **getLocationTrackingSessions**

parts	<b>parameter</b>
	element <b>csta:GetLocationTrackingSessions</b>
source	<message name="getLocationTrackingSessions">
	<part name="parameter" element="csta:GetLocationTrackingSessions"/>
	</message>



message **getLocationTrackingSessionsResponse**

parts	<b>parameter</b>
	element <b>csta:getLocationTrackingSessionsResponse</b>
source	<pre>&lt;message name="getLocationTrackingSessionsResponse"&gt;   &lt;part name="parameter" element="csta:getLocationTrackingSessionsResponse"/&gt; &lt;/message&gt;</pre>

message **locationSessionInfo**

parts	<b>parameter</b>
	element <b>csta:LocationSessionInfo</b>
source	<pre>&lt;message name="locationSessionInfo"&gt;   &lt;part name="parameter" element="csta:LocationSessionInfo"/&gt; &lt;/message&gt;</pre>

## 22.2 Operations

operations	<b>CSTA-Get-Location-Information</b>
	input <a href="#"><u>tns:getLocationInformation</u></a>
	output <a href="#"><u>tns:getLocationInformationResponse</u></a>
	fault <a href="#"><u>tns:negResponse</u></a>
	<b>CSTA-Set-Location-Information</b>
	input <a href="#"><u>tns:setLocationInformation</u></a>
	output <a href="#"><u>tns:setLocationInformationResponse</u></a>
	fault <a href="#"><u>tns:negResponse</u></a>
	<b>CSTA-Location-Tracking-Session-Resumed</b>
	input <a href="#"><u>tns:locationTrackingSessionResumed</u></a>
	output <a href="#"><u>tns:locationTrackingSessionResumedResponse</u></a>
	fault <a href="#"><u>tns:negResponse</u></a>
	<b>CSTA-Location-Tracking-Session-Suspended</b>
	input <a href="#"><u>tns:locationTrackingSessionSuspended</u></a>
	output <a href="#"><u>tns:locationTrackingSessionSuspendedResponse</u></a>
	fault <a href="#"><u>tns:negResponse</u></a>
	<b>CSTA-Resume-Location-Tracking-Session</b>
	input <a href="#"><u>tns:resumeLocationTrackingSession</u></a>
	output <a href="#"><u>tns:resumeLocationTrackingSessionResponse</u></a>
	fault <a href="#"><u>tns:negResponse</u></a>
	<b>CSTA-Location-Information-Report</b>

input [tns:locationInformationReport](#)  
 output [tns:locationInformationReportResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Start-Location-Tracking-Session**

input [tns:startLocationTrackingSession](#)  
 output [tns:startLocationTrackingSessionResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Stop-Location-Tracking-Session**

input [tns:stopLocationTrackingSession](#)  
 output [tns:stopLocationTrackingSessionResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Suspend-Location-Tracking-Session**

input [tns:suspendLocationTrackingSession](#)  
 output [tns:suspendLocationTrackingSessionResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Get-Location-Tracking-Capabilities**

input [tns:getLocationTrackingCapabilities](#)  
 output [tns:getLocationTrackingCapabilitiesResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Get-Location-Tracking-Sessions**

input [tns:getLocationTrackingSessions](#)  
 output [tns:getLocationTrackingSessionsResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Location-Session-Info**

source

```

  input tns:locationSessionInfo
  <operation name="CSTA-Get-Location-Information">
    <input message="tns:getLocationInformation"/>
    <output message="tns:getLocationInformationResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Set-Location-Information">
    <input message="tns:setLocationInformation"/>
    <output message="tns:setLocationInformationResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Location-Tracking-Session-Resumed">
    <output message="tns:locationTrackingSessionResumed"/>
    <input message="tns:locationTrackingSessionResumedResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>

```

```

</operation>
<operation name="CSTA-Location-Tracking-Session-Suspended">
  <input message="tns:locationTrackingSessionSuspended"/>
  <output message="tns:locationTrackingSessionSuspendedResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Resume-Location-Tracking-Session">
  <input message="tns:resumeLocationTrackingSession"/>
  <output message="tns:resumeLocationTrackingSessionResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Location-Information-Report">
  <input message="tns:locationInformationReport"/>
  <output message="tns:locationInformationReportResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Start-Location-Tracking-Session">
  <input message="tns:startLocationTrackingSession"/>
  <output message="tns:startLocationTrackingSessionResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Stop-Location-Tracking-Session">
  <input message="tns:stopLocationTrackingSession"/>
  <output message="tns:stopLocationTrackingSessionResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Suspend-Location-Tracking-Session">
  <input message="tns:suspendLocationTrackingSession"/>
  <output message="tns:suspendLocationTrackingSessionResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Location-Tracking-Capabilities">
  <input message="tns:getLocationTrackingCapabilities"/>
  <output message="tns:getLocationTrackingCapabilitiesResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Get-Location-Tracking-Sessions">
  <input message="tns:getLocationTrackingSessions"/>
  <output message="tns:getLocationTrackingSessionsResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Location-Session-Info">
  <input message="tns:locationSessionInfo"/>
</operation>

```

## 23 Vendor Specific Features - Messages and Operations

This Clause specifies the WSDL messages and operations for the Vendor Specific features.

### 23.1 Messages

#### message **escapeRegister**

parts	<b>parameter</b>
	element <b>csta:EscapeRegister</b>
source	<code>&lt;message name="escapeRegister"&gt;</code>
	<code>&lt;part name="parameter" element="csta:EscapeRegister"/&gt;</code>
	<code>&lt;/message&gt;</code>

#### message **escapeRegisterResponse**

parts	<b>parameter</b>
	element <b>csta:EscapeRegisterResponse</b>
source	<code>&lt;message name="escapeRegisterResponse"&gt;</code>
	<code>&lt;part name="parameter" element="csta:EscapeRegisterResponse"/&gt;</code>
	<code>&lt;/message&gt;</code>

#### message **escapeRegisterAbort**

parts	<b>parameter</b>
	element <b>csta:EscapeRegisterAbort</b>
source	<code>&lt;message name="escapeRegisterAbort"&gt;</code>
	<code>&lt;part name="parameter" element="csta:EscapeRegisterAbort"/&gt;</code>
	<code>&lt;/message&gt;</code>

#### message **escapeRegisterCancel**

parts	<b>parameter</b>
	element <b>csta:EscapeRegisterCancel</b>
source	<code>&lt;message name="escapeRegisterCancel"&gt;</code>
	<code>&lt;part name="parameter" element="csta:EscapeRegisterCancel"/&gt;</code>
	<code>&lt;/message&gt;</code>

#### message **escapeRegisterCancelResponse**

parts	<b>parameter</b>
	element <b>csta:EscapeRegisterCancelResponse</b>
source	<code>&lt;message name="escapeRegisterCancelResponse"&gt;</code>
	<code>&lt;part name="parameter" element="csta:EscapeRegisterCancelResponse"/&gt;</code>
	<code>&lt;/message&gt;</code>

**message escape**

parts	<b>parameter</b>
	element <b>csta:Escape</b>
source	<pre>&lt;message name="escape"&gt;   &lt;part name="parameter" element="csta:Escape"/&gt; &lt;/message&gt;</pre>

**message escapeResponse**

parts	<b>parameter</b>
	element <b>csta:EscapeResponse</b>
source	<pre>&lt;message name="escapeResponse"&gt;   &lt;part name="parameter" element="csta:EscapeResponse"/&gt; &lt;/message&gt;</pre>

**message privateDataVersionSelection**

parts	<b>parameter</b>
	element <b>csta:PrivateDataVersionSelection</b>
source	<pre>&lt;message name="privateDataVersionSelection"&gt;   &lt;part name="parameter" element="csta:PrivateDataVersionSelection"/&gt; &lt;/message&gt;</pre>

**message privateDataVersionSelectionResponse**

parts	<b>parameter</b>
	element <b>csta:PrivateDataVersionSelectionResponse</b>
source	<pre>&lt;message name="privateDataVersionSelectionResponse"&gt;   &lt;part name="parameter" element="csta:PrivateDataVersionSelectionResponse"/&gt; &lt;/message&gt;</pre>

**message privateEvent**

parts	<b>parameter</b>
	element <b>csta:PrivateEvent</b>
source	<pre>&lt;message name="privateEvent"&gt;   &lt;part name="parameter" element="csta:PrivateEvent"/&gt; &lt;/message&gt;</pre>

## 23.2 Operations

operations

### CSTA-EscapeRegister

input [tns:escapeRegister](#)  
 output [tns:escapeRegisterResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Escape-Register-Abort

input [tns:escapeRegisterAbort](#)

### CSTA-Escape-Register-Cancel

input [tns:escapeRegisterCancel](#)  
 output [tns:escapeRegisterCancelResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Escape-as-input

input [tns:escape](#)  
 output [tns:escapeResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Escape-as-output

input [tns:escape](#)  
 output [tns:escapeResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Private-Data-Version-Selection

input [tns:privateDataVersionSelection](#)  
 output [tns:privateDataVersionSelectionResponse](#)  
 fault [tns:negResponse](#)

### CSTA-Private-Event

source

```

  input tns:privateEvent
<operation name="CSTA-EscapeRegister">
  <input message="tns:escapeRegister"/>
  <output message="tns:escapeRegisterResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Escape-Register-Abort">
  <input message="tns:escapeRegisterAbort"/>
</operation>
<operation name="CSTA-Escape-Register-Cancel">
  <input message="tns:escapeRegisterCancel"/>
  <output message="tns:escapeRegisterCancelResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
```

```

<operation name="CSTA-Escape-as-input">
  <input message="tns:escape"/>
  <output message="tns:escapeResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Escape-as-output">
  <input message="tns:escape"/>
  <output message="tns:escapeResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Private-Data-Version-Selection">
  <input message="tns:privateDataVersionSelection"/>
  <output message="tns:privateDataVersionSelectionResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Private-Event">
  <input message="tns:privateEvent"/>
</operation>

```

## 24 Basic Telephony Profile – Port Type Pair

This Clause specifies the CSTA-Basic-Telephony-Profile portType and CF-CSTA-Basic-Telephony-Profile portType. This portType pair specifies the required CSTA WSDL operations to support the CSTA Basic Telephony profile as specified in ISO/IEC 18051 2.1.3.1.

Implementations may support other operations in addition to the operations specified in this port type pair.

porttype **CSTA-Basic-Telephony-Profile**

operations	<b>CSTA-Get-Switching-Function-Capabilities</b> input <a href="#">tns:getSwitchingFunctionCaps</a> output <a href="#">tns:getSwitchingFunctionCapsResponse</a> fault <a href="#">tns:negResponse</a>  <b>CSTA-Answer-Call</b> input <a href="#">tns:answerCall</a> output <a href="#">tns:answerCallResponse</a> fault <a href="#">tns:negResponse</a>  <b>CSTA-Clear-Connection</b> input <a href="#">tns:clearConnection</a> output <a href="#">tns:clearConnectionResponse</a> fault <a href="#">tns:negResponse</a>
------------	--

**CSTA-Make-Call**

input [tns:makeCall](#)  
 output [tns:makeCallResponse](#)  
 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](#)

source `<portType name="CSTA-Basic-Telephony-Profile">`  
`<operation name="CSTA-Get-Switching-Function-Capabilities">`  
`<input message="tns:getSwitchingFunctionCaps"/>`  
`<output message="tns:getSwitchingFunctionCapsResponse"/>`  
`<fault name="FaultName" message="tns:negResponse"/>`  
`</operation>`  
`<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>`  
`<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>`  
`</portType>`



porttype **CF-CSTA-Basic-Telephony-Profile**

operations	<b>CSTA-Connection-Cleared-event</b>
	input <a href="#">tns:connectionClearedEvent</a>
	<b>CSTA-Delivered-event</b>
	input <a href="#">tns:deliveredEvent</a>
	<b>CSTA-Established-event</b>
	input <a href="#">tns:establishedEvent</a>
	<b>CSTA-Failed-event</b>
source	input <a href="#">tns:failedEvent</a>
	<b>CSTA-Network-Reached-event</b>
	input <a href="#">tns:networkReachedEvent</a>
	<b>CSTA-Originated-event</b>
	input <a href="#">tns:originatedEvent</a>
	<b>CSTA-Service-Initiated-event</b>
	input <a href="#">tns:serviceInitiatedEvent</a>
	<pre> &lt;portType name=" CF-CSTA-Basic-Telephony-Profile"&gt;   &lt;operation name="CSTA-Connection-Cleared-event"&gt;     &lt;input message="tns:connectionClearedEvent"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Delivered-event"&gt;     &lt;input message="tns:deliveredEvent"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Established-event"&gt;     &lt;input message="tns:establishedEvent"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Failed-event"&gt;     &lt;input message="tns:failedEvent"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Network-Reached-event"&gt;     &lt;input message="tns:networkReachedEvent"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Originated-event"&gt;     &lt;input message="tns:originatedEvent"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Service-Initiated-event"&gt;     &lt;input message="tns:serviceInitiatedEvent"/&gt;   &lt;/operation&gt; &lt;/portType&gt; </pre>

## 25 Routeing Profile - Port Type Pair

This Clause specifies the CSTA-Routeing-Profile portType and CF-CSTA-Routeing-Profile portType. This portType pair specifies the required CSTA WSDL operations to support the CSTA Routeing profile as specified in ISO/IEC 18051 2.1.3.2.

Implementations may support other operations in addition to the operations specified in this port type pair.

### porttype **CSTA-Routeing-Profile**

operations	<b>CSTA-Route-Select</b>
	input <a href="#">tns:routeSelect</a>
source	<pre> &lt;portType name="CSTA-Routeing-Profile"&gt;   &lt;operation name="CSTA-Route-Select"&gt;     &lt;input message="tns:routeSelect"/&gt;   &lt;/operation&gt; &lt;/portType&gt; </pre>

### porttype **CF-CSTA-Routeing-Profile**

operations	<b>CSTA-Route-Request</b>
	input <a href="#">tns:routeRequest</a>
	<b>CSTA-Route-End-as-output</b>
	input <a href="#">tns:routeEnd</a>
source	<pre> &lt;portType name="CF-CSTA-Routeing-Profile"&gt;   &lt;operation name="CSTA-Route-Request"&gt;     &lt;input message="tns:routeRequest"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Route-End-as-output"&gt;     &lt;input message="tns:routeEnd"/&gt;   &lt;/operation&gt; &lt;/portType&gt; </pre>

## 26 Level 1a Voice Browser Profile - Port Type Pair

This Clause specifies the CSTA-Level-1a-Voice-Browser-Profile portType and CF- CSTA-Level-1a-Voice-Browser-Profile portType. This portType pair specifies the required CSTA WSDL operations to support the CSTA Level 1a Voice Browser profile as specified in ISO/IEC 18051 2.1.3.3.

Implementations may support other operations in addition to the operations specified in this port type pair.

porttype **CSTA-Level-1a-Voice-Browser-Profile**

operations	<p><b>CSTA-Answer-Call</b></p> <p>input    <a href="#">tns:answerCall</a></p> <p>output   <a href="#">tns:answerCallResponse</a></p> <p>fault    <a href="#">tns:negResponse</a></p> <p><b>CSTA-Clear-Connection</b></p> <p>input    <a href="#">tns:clearConnection</a></p> <p>output   <a href="#">tns:clearConnectionResponse</a></p> <p>fault    <a href="#">tns:negResponse</a></p> <p><b>CSTA-Single-Step-Transfer-Call</b></p> <p>input    <a href="#">tns:singleStepTransferCall</a></p> <p>output   <a href="#">tns:singleStepTransferCallResponse</a></p> <p>fault    <a href="#">tns:negResponse</a></p> <p><b>CSTA-Monitor-Start</b></p> <p>input    <a href="#">tns:monitorStart</a></p> <p>output   <a href="#">tns:monitorStartResponse</a></p> <p>fault    <a href="#">tns:negResponse</a></p> <p><b>CSTA-Monitor-Stop-as-input</b></p> <p>input    <a href="#">tns:monitorStop</a></p> <p>output   <a href="#">tns:monitorStopResponse</a></p> <p>fault    <a href="#">tns:negResponse</a></p>
source	<pre> &lt;portType name="CSTA-Level-1a-Voice-Browser-Profile"&gt;   &lt;operation name="CSTA-Answer-Call"&gt;     &lt;input message="tns:answerCall"/&gt;     &lt;output message="tns:answerCallResponse"/&gt;     &lt;fault name="FaultName" message="tns:negResponse"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Clear-Connection"&gt;     &lt;input message="tns:clearConnection"/&gt;     &lt;output message="tns:clearConnectionResponse"/&gt;     &lt;fault name="FaultName" message="tns:negResponse"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Single-Step-Transfer-Call"&gt;     &lt;input message="tns:singleStepTransferCall"/&gt;     &lt;output message="tns:singleStepTransferCallResponse"/&gt; </pre>

```

    <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>
</portType>

```

### porttype CF-CSTA-Level-1a-Voice-Browser-Profile

operations	<b>CSTA-Connection-Cleared-event</b> input <a href="#">tns:connectionClearedEvent</a>
	<b>CSTA-Failed-event</b> input <a href="#">tns:failedEvent</a>
	<b>CSTA-Delivered-event</b> input <a href="#">tns:deliveredEvent</a>
	<b>CSTA-Transferred-event</b> input <a href="#">tns:transferredEvent</a>
	<b>CSTA-Established-event</b> input <a href="#">tns:establishedEvent</a>
source	<pre> &lt;portType name="CF-CSTA-Level-1a-Voice-Browser-Profile"&gt;   &lt;operation name="CSTA-Connection-Cleared-event"&gt;     &lt;input message="tns:connectionClearedEvent"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Failed-event"&gt;     &lt;input message="tns:failedEvent"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Delivered-event"&gt;     &lt;input message="tns:deliveredEvent"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Transferred-event"&gt;     &lt;input message="tns:transferredEvent"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Established-event"&gt;     &lt;input message="tns:establishedEvent"/&gt;   &lt;/operation&gt; &lt;/portType&gt; </pre>

## 27 Level 1b Voice Browser Profile - Port Type Pair

This Clause specifies the CSTA-Level-1b-Voice-Browser-Profile portType and CSTA-Level-1b-Voice-Browser-Profile portType. This portType pair specifies the required CSTA WSDL operations to support the CSTA Level 1b Voice Browser profile as specified in ISO/IEC 18051 2.1.3.4.

Implementations may support other operations in addition to the operations specified in this port type pair.

porttype **CSTA-Level-1b-Voice-Browser-Profile**

operations	<p><b>CSTA-Answer-Call</b></p> <p>input    <a href="#">tns:answerCall</a></p> <p>output   <a href="#">tns:answerCallResponse</a></p> <p>fault    <a href="#">tns:negResponse</a></p> <p><b>CSTA-Clear-Connection</b></p> <p>input    <a href="#">tns:clearConnection</a></p> <p>output   <a href="#">tns:clearConnectionResponse</a></p> <p>fault    <a href="#">tns:negResponse</a></p> <p><b>CSTA-Deflect-Call</b></p> <p>input    <a href="#">tns:deflectCall</a></p> <p>output   <a href="#">tns:deflectCallResponse</a></p> <p>fault    <a href="#">tns:negResponse</a></p> <p><b>CSTA-Monitor-Start</b></p> <p>input    <a href="#">tns:monitorStart</a></p> <p>output   <a href="#">tns:monitorStartResponse</a></p> <p>fault    <a href="#">tns:negResponse</a></p> <p><b>CSTA-Monitor-Stop-as-input</b></p> <p>input    <a href="#">tns:monitorStop</a></p> <p>output   <a href="#">tns:monitorStopResponse</a></p> <p>fault    <a href="#">tns:negResponse</a></p>
source	<pre> &lt;portType name="CSTA-Level-1b-Voice-Browser-Profile"&gt;   &lt;operation name="CSTA-Answer-Call"&gt;     &lt;input message="tns:answerCall"/&gt;     &lt;output message="tns:answerCallResponse"/&gt;     &lt;fault name="FaultName" message="tns:negResponse"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Clear-Connection"&gt;     &lt;input message="tns:clearConnection"/&gt;     &lt;output message="tns:clearConnectionResponse"/&gt;     &lt;fault name="FaultName" message="tns:negResponse"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Deflect-Call"&gt;     &lt;input message="tns:deflectCall"/&gt;     &lt;output message="tns:deflectCallResponse"/&gt; </pre>

```

    <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>
</portType>

```

### porttype CF-CSTA-Level-1b-Voice-Browser-Profile

operations	<b>CSTA-Connection-Cleared-event</b> input <a href="#">tns:connectionClearedEvent</a>
	<b>CSTA-Failed-event</b> input <a href="#">tns:failedEvent</a>
	<b>CSTA-Delivered-event</b> input <a href="#">tns:deliveredEvent</a>
	<b>CSTA-Diverted-event</b> input <a href="#">tns:divertedEvent</a>
	<b>CSTA-Established-event</b> input <a href="#">tns:establishedEvent</a>
source	<pre> &lt;portType name="CF-CSTA-Level-1b-Voice-Browser-Profile"&gt;   &lt;operation name="CSTA-Connection-Cleared-event"&gt;     &lt;input message="tns:connectionClearedEvent"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Failed-event"&gt;     &lt;input message="tns:failedEvent"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Delivered-event"&gt;     &lt;input message="tns:deliveredEvent"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Diverted-event"&gt;     &lt;input message="tns:divertedEvent"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Established-event"&gt;     &lt;input message="tns:establishedEvent"/&gt;   &lt;/operation&gt; &lt;/portType&gt; </pre>

## 28 Level 2a Voice Browser Profile - Port Type Pair

This Clause specifies the CSTA-Level-2a-Voice-Browser-Profile portType and CF-CSTA-Level-2a-Voice-Browser-Profile portType. This portType pair specifies the required CSTA WSDL operations to support the CSTA Level 2a Voice Browser profile as specified in ISO/IEC 18051 2.1.3.5.

Implementations may support other operations in addition to the operations specified in this port type pair.

porttype **CSTA-Level-2a-Voice-Browser-Profile**

operations	<b>CSTA-Answer-Call</b>
	input <a href="#">tns:answerCall</a> output <a href="#">tns:answerCallResponse</a> fault <a href="#">tns:negResponse</a>
	<b>CSTA-Clear-Connection</b>
	input <a href="#">tns:clearConnection</a> output <a href="#">tns:clearConnectionResponse</a> fault <a href="#">tns:negResponse</a>
	<b>CSTA-Make-Call</b>
	input <a href="#">tns:makeCall</a> output <a href="#">tns:makeCallResponse</a> fault <a href="#">tns:negResponse</a>
	<b>CSTA-Single-Step-Transfer-Call</b>
	input <a href="#">tns:singleStepTransferCall</a> output <a href="#">tns:singleStepTransferCallResponse</a> fault <a href="#">tns:negResponse</a>
	<b>CSTA-Monitor-Start</b>
	input <a href="#">tns:monitorStart</a> output <a href="#">tns:monitorStartResponse</a> fault <a href="#">tns:negResponse</a>
	<b>CSTA-Monitor-Stop-as-input</b>
	input <a href="#">tns:monitorStop</a> output <a href="#">tns:monitorStopResponse</a> fault <a href="#">tns:negResponse</a>
source	<pre> &lt;portType name="CSTA-Level-2a-Voice-Browser-Profile"&gt;   &lt;operation name="CSTA-Answer-Call"&gt;     &lt;input message="tns:answerCall"/&gt;     &lt;output message="tns:answerCallResponse"/&gt;     &lt;fault name="FaultName" message="tns:negResponse"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Clear-Connection"&gt;     &lt;input message="tns:clearConnection"/&gt;     &lt;output message="tns:clearConnectionResponse"/&gt; </pre>

```

    <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>
  <operation name="CSTA-Single-Step-Transfer-Call">
    <input message="tns:singleStepTransferCall"/>
    <output message="tns:singleStepTransferCallResponse"/>
    <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>
</portType>

```

## porttype CF-CSTA-Level-2a-Voice-Browser-Profile

operations

### CSTA-Connection-Cleared-event

input [tns:connectionClearedEvent](#)

### CSTA-Delivered-event

input [tns:deliveredEvent](#)

### CSTA-Established-event

input [tns:establishedEvent](#)

### CSTA-Failed-event

input [tns:failedEvent](#)

### CSTA-Network-Reached-event

input [tns:networkReachedEvent](#)

### CSTA-Originated-event

input [tns:originatedEvent](#)

### CSTA-Transferred-event

input [tns:transferredEvent](#)



```

source <portType name="CF-CSTA-Level-2a-Voice-Browser-Profile">
  <operation name="CSTA-Connection-Cleared-event">
    <input message="tns:connectionClearedEvent"/>
  </operation>
  <operation name="CSTA-Delivered-event">
    <input message="tns:deliveredEvent"/>
  </operation>
  <operation name="CSTA-Established-event">
    <input message="tns:establishedEvent"/>
  </operation>
  <operation name="CSTA-Failed-event">
    <input message="tns:failedEvent"/>
  </operation>
  <operation name="CSTA-Network-Reached-event">
    <input message="tns:networkReachedEvent"/>
  </operation>
  <operation name="CSTA-Originated-event">
    <input message="tns:originatedEvent"/>
  </operation>
  <operation name="CSTA-Transferred-event">
    <input message="tns:transferredEvent"/>
  </operation>
</portType>

```

## 29 Level 2b Voice Browser Profile - Port Type Pair

This Clause specifies the CSTA-Level-2b-Voice-Browser-Profile portType and CF- CSTA-Level-2b-Voice-Browser-Profile portType. This portType pair specifies the required CSTA WSDL operations to support the CSTA Level 2b Voice Browser profile as specified in ISO/IEC 18051 2.1.3.5.

Implementations may support other operations in addition to the operations specified in this port type pair.

porttype **CSTA-Level-2b-Voice-Browser-Profile**

operations	<b>CSTA-Answer-Call</b>  input <a href="#">tns:answerCall</a> output <a href="#">tns:answerCallResponse</a> fault <a href="#">tns:negResponse</a>  <b>CSTA-Clear-Connection</b>  input <a href="#">tns:clearConnection</a> output <a href="#">tns:clearConnectionResponse</a> fault <a href="#">tns:negResponse</a>  <b>CSTA-Make-Call</b>  input <a href="#">tns:makeCall</a> output <a href="#">tns:makeCallResponse</a> fault <a href="#">tns:negResponse</a>
------------	--

**CSTA-Deflect-Call**

input [tns:deflectCall](#)  
 output [tns:deflectCallResponse](#)  
 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](#)

source <portType name="CSTA-Level-2b-Voice-Browser-Profile">  
 <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>  
 <operation name="CSTA-Deflect-Call">  
 <input message="tns:deflectCall"/>  
 <output message="tns:deflectCallResponse"/>  
 <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>  
 </portType>

porttype **CF-CSTA-Level-2b-Voice-Browser-Profile**operations **CSTA-Connection-Cleared-event**input [tns:connectionClearedEvent](#)**CSTA-Delivered-event**input [tns:deliveredEvent](#)**CSTA-Established-event**input [tns:establishedEvent](#)**CSTA-Failed-event**input [tns:failedEvent](#)**CSTA-Network-Reached-event**input [tns:networkReachedEvent](#)**CSTA-Originated-event**input [tns:originatedEvent](#)**CSTA-Diverted-event**input [tns:divertedEvent](#)

```

source <portType name="CF-CSTA-Level-2b-Voice-Browser-Profile">
  <operation name="CSTA-Connection-Cleared-event">
    <input message="tns:connectionClearedEvent"/>
  </operation>
  <operation name="CSTA-Delivered-event">
    <input message="tns:deliveredEvent"/>
  </operation>
  <operation name="CSTA-Established-event">
    <input message="tns:establishedEvent"/>
  </operation>
  <operation name="CSTA-Failed-event">
    <input message="tns:failedEvent"/>
  </operation>
  <operation name="CSTA-Network-Reached-event">
    <input message="tns:networkReachedEvent"/>
  </operation>
  <operation name="CSTA-Originated-event">
    <input message="tns:originatedEvent"/>
  </operation>
  <operation name="CSTA-Diverted-event">
    <input message="tns:divertedEvent"/>
  </operation>
</portType>

```

### 30 Minimal uaCSTA Call Control Profile – Port Type

This Clause specifies the CSTA-Minimal-uaCSTA-Call-Control-Profile portType. This portType specifies the required CSTA WSDL operations to support the CSTA Minimal uaCSTA Call Control Profile as specified in ISO/IEC 18051 2.1.3.6.

Implementations may support other operations in addition to the operations specified in this port type.

porttype **CSTA-Minimal-uaCSTA-Call-Control-Profile**

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-Deflect-Call

input [tns:deflectCall](#)  
 output [tns:deflectCallResponse](#)  
 fault [tns:negResponse](#)

#### CSTA-Hold-Call

input [tns:holdCall](#)  
 output [tns:holdCallResponse](#)  
 fault [tns:negResponse](#)

#### CSTA-Make-Call

input [tns:makeCall](#)  
 output [tns:makeCallResponse](#)  
 fault [tns:negResponse](#)

#### CSTA-Retrieve-Call

input [tns:retrieveCall](#)  
 output [tns:retrieveCallResponse](#)  
 fault [tns:negResponse](#)

#### CSTA-Single-Step-Transfer-Call

input [tns:singleStepTransferCall](#)  
 output [tns:singleStepTransferCallResponse](#)  
 fault [tns:negResponse](#)

```

source <portType name="CSTA-Minimal-uaCSTA-Call-Control-Profile">
  <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-Deflect-Call">
    <input message="tns:deflectCall"/>
    <output message="tns:deflectCallResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Hold-Call">
    <input message="tns:holdCall"/>
    <output message="tns:holdCallResponse"/>
    <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>
  <operation name="CSTA-Retrieve-Call">
    <input message="tns:retrieveCall"/>
    <output message="tns:retrieveCallResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Single-Step-Transfer-Call">
    <input message="tns:singleStepTransferCall"/>
    <output message="tns:singleStepTransferCallResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
</portType>

```

### 31 Basic uaCSTA Call Control Profile – Port Type Pair

This Clause specifies the CSTA-Basic-uaCSTA-Call-Control-Profile portType and CF-CSTA-Basic-uaCSTA-Call-Control-Profile port. This portType pair specifies the required CSTA WSDL operations to support the CSTA Basic uaCSTA Call Control Profile as specified in ISO/IEC 18051 2.1.3.7.

Implementations may support other operations in addition to the operations specified in this port type pair.

porttype **CSTA-Basic-uaCSTA-Call-Control-Profile**

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-Deflect-Call

input [tns:deflectCall](#)  
 output [tns:deflectCallResponse](#)  
 fault [tns:negResponse](#)

#### CSTA-Hold-Call

input [tns:holdCall](#)  
 output [tns:holdCallResponse](#)  
 fault [tns:negResponse](#)

#### CSTA-Make-Call

input [tns:makeCall](#)  
 output [tns:makeCallResponse](#)  
 fault [tns:negResponse](#)

#### CSTA-Retrieve-Call

input [tns:retrieveCall](#)  
 output [tns:retrieveCallResponse](#)  
 fault [tns:negResponse](#)

#### CSTA-Single-Step-Transfer-Call

input [tns:singleStepTransferCall](#)  
 output [tns:singleStepTransferCallResponse](#)  
 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](#)

```

source <portType name="CSTA-Basic-uaCSTA-Call-Control-Profile">
  <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-Deflect-Call">
    <input message="tns:deflectCall"/>
    <output message="tns:deflectCallResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Hold-Call">
    <input message="tns:holdCall"/>
    <output message="tns:holdCallResponse"/>
    <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>
  <operation name="CSTA-Retrieve-Call">
    <input message="tns:retrieveCall"/>
    <output message="tns:retrieveCallResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
  <operation name="CSTA-Single-Step-Transfer-Call">
    <input message="tns:singleStepTransferCall"/>
    <output message="tns:singleStepTransferCallResponse"/>
    <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>
<operation name="CSTA-Monitor-Stop-as-input">
  <input message="tns:monitorStop"/>
  <output message="tns:monitorStopResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
</portType>

```

#### porttype CF-CSTA-Basic-uaCSTA-Call-Control-Profile

operations	<b>CSTA-Connection-Cleared-event</b> input <a href="#">tns:connectionClearedEvent</a>
	<b>CSTA-Delivered-event</b> input <a href="#">tns:deliveredEvent</a>
	<b>CSTA-Diverted-event</b> input <a href="#">tns:divertedEvent</a>
	<b>CSTA-Established-event</b> input <a href="#">tns:establishedEvent</a>
	<b>CSTA-Failed-event</b> input <a href="#">tns:failedEvent</a>
	<b>CSTA-Held-event</b> input <a href="#">tns:heldEvent</a>
	<b>CSTA-Network-Reached-event</b> input <a href="#">tns:networkReachedEvent</a>
	<b>CSTA-Retrieved-event</b> input <a href="#">tns:retrievedEvent</a>
	<b>CSTA-Service-Initiated-event</b> input <a href="#">tns:serviceInitiatedEvent</a>
	<b>CSTA-Transferred-event</b> input <a href="#">tns:transferredEvent</a>
source	<pre> &lt;portType name="CF-CSTA-Basic-uaCSTA-Call-Control-Profile"&gt;   &lt;operation name="CSTA-Connection-Cleared-event"&gt;     &lt;input message="tns:connectionClearedEvent"/&gt;   &lt;/operation&gt;   &lt;operation name="CSTA-Delivered-event"&gt; </pre>



```

    <input message="tns:deliveredEvent"/>
  </operation>
  <operation name="CSTA-Diverted-event">
    <input message="tns:divertedEvent"/>
  </operation>
  <operation name="CSTA-Established-event">
    <input message="tns:establishedEvent"/>
  </operation>
  <operation name="CSTA-Failed-event">
    <input message="tns:failedEvent"/>
  </operation>
  <operation name="CSTA-Held-event">
    <input message="tns:heldEvent"/>
  </operation>
  <operation name="CSTA-Network-Reached-event">
    <input message="tns:networkReachedEvent"/>
  </operation>
  <operation name="CSTA-Retrieved-event">
    <input message="tns:retrievedEvent"/>
  </operation>
  <operation name="CSTA-Service-Initiated-event">
    <input message="tns:serviceInitiatedEvent"/>
  </operation>
  <operation name="CSTA-Transferred-event">
    <input message="tns:transferredEvent"/>
  </operation>
</portType>

```

### 32 Advanced uaCSTA Call Control Profile – Port Type Pair

This Clause specifies the CSTA-Advanced-uaCSTA-Call-Control-Profile portType and CF- CSTA-Advanced-uaCSTA-Call-Control-Profile portType. This portType pair specifies the required CSTA WSDL operations to support the CSTA Advanced uaCSTA Call Control Profile as specified in ISO/IEC 18051 2.1.3.8.

Implementations may support other operations in addition to the operations specified in this port type pair.

#### porttype **CSTA-Advanced-uaCSTA-Call-Control-Profile**

operations	<b>CSTA-Alternate-Call</b>  input <a href="#"><u>tns:alternateCall</u></a> output <a href="#"><u>tns:alternateCallResponse</u></a> fault <a href="#"><u>tns:negResponse</u></a>  <b>CSTA-Answer-Call</b>  input <a href="#"><u>tns:answerCall</u></a> output <a href="#"><u>tns:answerCallResponse</u></a> fault <a href="#"><u>tns:negResponse</u></a>
------------	---

**CSTA-Clear-Connection**

input [tns:clearConnection](#)  
 output [tns:clearConnectionResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Consultation-Call**

input [tns:consultationCall](#)  
 output [tns:consultationCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Deflect-Call**

input [tns:deflectCall](#)  
 output [tns:deflectCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Hold-Call**

input [tns:holdCall](#)  
 output [tns:holdCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Make-Call**

input [tns:makeCall](#)  
 output [tns:makeCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Reconnect-Call**

input [tns:reconnectCall](#)  
 output [tns:reconnectCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Retrieve-Call**

input [tns:retrieveCall](#)  
 output [tns:retrieveCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Single-Step-Transfer-Call**

input [tns:singleStepTransferCall](#)  
 output [tns:singleStepTransferCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Transfer-Call**

input [tns:transferCall](#)  
 output [tns:transferCallResponse](#)  
 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-Get-Switching-Function-Capabilities**

input [tns:getSwitchingFunctionCaps](#)  
 output [tns:getSwitchingFunctionCapsResponse](#)  
 fault [tns:negResponse](#)

source <portType name="CSTA-Advanced-uaCSTA-Call-Control-Profile">  
 <operation name="CSTA-Alternate-Call">  
 <input message="tns:alternateCall"/>  
 <output message="tns:alternateCallResponse"/>  
 <fault name="FaultName" message="tns:negResponse"/>  
 </operation>  
 <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-Consultation-Call">  
 <input message="tns:consultationCall"/>  
 <output message="tns:consultationCallResponse"/>  
 <fault name="FaultName" message="tns:negResponse"/>  
 </operation>  
 <operation name="CSTA-Deflect-Call">  
 <input message="tns:deflectCall"/>  
 <output message="tns:deflectCallResponse"/>  
 <fault name="FaultName" message="tns:negResponse"/>  
 </operation>  
 <operation name="CSTA-Hold-Call">  
 <input message="tns:holdCall"/>  
 <output message="tns:holdCallResponse"/>  
 <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>
<operation name="CSTA-Reconnect-Call">
  <input message="tns:reconnectCall"/>
  <output message="tns:reconnectCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Retrieve-Call">
  <input message="tns:retrieveCall"/>
  <output message="tns:retrieveCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Single-Step-Transfer-Call">
  <input message="tns:singleStepTransferCall"/>
  <output message="tns:singleStepTransferCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Transfer-Call">
  <input message="tns:transferCall"/>
  <output message="tns:transferCallResponse"/>
  <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-Get-Switching-Function-Capabilities">
  <input message="tns:getSwitchingFunctionCaps"/>
  <output message="tns:getSwitchingFunctionCapsResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
</portType>

```

porttype **CF-CSTA-Advanced-uaCSTA-Call-Control-Profile**

operations

**CSTA-Connection-Cleared-event**input [tns:connectionClearedEvent](#)**CSTA-Delivered-event**input [tns:deliveredEvent](#)**CSTA-Diverted-event**input [tns:divertedEvent](#)**CSTA-Established-event**input [tns:establishedEvent](#)**CSTA-Failed-event**input [tns:failedEvent](#)**CSTA-Held-event**input [tns:heldEvent](#)**CSTA-Network-Reached-event**input [tns:networkReachedEvent](#)**CSTA-Originated-event**input [tns:originatedEvent](#)**CSTA-Retrieved-event**input [tns:retrievedEvent](#)**CSTA-Service-Initiated-event**input [tns:serviceInitiatedEvent](#)**CSTA-Transferred-event**input [tns:transferredEvent](#)

source

```

<portType name="CF-CSTA-Advanced-uaCSTA-Call-Control-Profile">
  <operation name="CSTA-Delivered-event">
    <input message="tns:deliveredEvent"/>
  </operation>
  <operation name="CSTA-Diverted-event">
    <input message="tns:divertedEvent"/>
  </operation>
  <operation name="CSTA-Established-event">
    <input message="tns:establishedEvent"/>
  </operation>

```

```

<operation name="CSTA-Failed-event">
  <input message="tns:failedEvent"/>
</operation>
<operation name="CSTA-Held-event">
  <input message="tns:heldEvent"/>
</operation>
<operation name="CSTA-Network-Reached-event">
  <input message="tns:networkReachedEvent"/>
</operation>
<operation name="CSTA-Originated-event">
  <input message="tns:originatedEvent"/>
</operation>
<operation name="CSTA-Retrieved-event">
  <input message="tns:retrievedEvent"/>
</operation>
<operation name="CSTA-Service-Initiated-event">
  <input message="tns:serviceInitiatedEvent"/>
</operation>
<operation name="CSTA-Transferred-event">
  <input message="tns:transferredEvent"/>
</operation>
</portType>

```

### 33 Conferencing uaCSTA with Basic Call Control Profile – Port Type Pair

This Clause specifies the CSTA-Conferencing-uaCSTA-Basic-Call-Control-Profile portType and CF- CSTA-Conferencing-uaCSTA-Basic-Call-Control-Profile portType. This portType pair specifies the required CSTA WSDL operations to support the CSTA Conferencing uaCSTA Profile with the Basic uaCSTA Call Control Profile as specified in ISO/IEC 18051 2.1.3.9.

Implementations may support other operations in addition to the operations specified in this port type pair.

#### porttype CSTA-Conferencing-uaCSTA-Basic-Call-Control-Profile

operations	<b>CSTA-Answer-Call</b>  input <a href="#"><u>tns:answerCall</u></a> output <a href="#"><u>tns:answerCallResponse</u></a> fault <a href="#"><u>tns:negResponse</u></a>  <b>CSTA-Clear-Connection</b>  input <a href="#"><u>tns:clearConnection</u></a> output <a href="#"><u>tns:clearConnectionResponse</u></a> fault <a href="#"><u>tns:negResponse</u></a>
------------	---

**CSTA-Deflect-Call**

input [tns:deflectCall](#)  
 output [tns:deflectCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Hold-Call**

input [tns:holdCall](#)  
 output [tns:holdCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Make-Call**

input [tns:makeCall](#)  
 output [tns:makeCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Retrieve-Call**

input [tns:retrieveCall](#)  
 output [tns:retrieveCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Single-Step-Transfer-Call**

input [tns:singleStepTransferCall](#)  
 output [tns:singleStepTransferCallResponse](#)  
 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](#)

source `<portType name="CSTA-Conferencing-uaCSTA-Basic-Call-Control-Profile">`  
`<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-Deflect-Call">
  <input message="tns:deflectCall"/>
  <output message="tns:deflectCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Hold-Call">
  <input message="tns:holdCall"/>
  <output message="tns:holdCallResponse"/>
  <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>
<operation name="CSTA-Retrieve-Call">
  <input message="tns:retrieveCall"/>
  <output message="tns:retrieveCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Single-Step-Transfer-Call">
  <input message="tns:singleStepTransferCall"/>
  <output message="tns:singleStepTransferCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Connection-Cleared-event">
  <output message="tns:connectionClearedEvent"/>
</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>
</portType>

```



porttype **CF-CSTA-Conferencing-uaCSTA-Basic-Call-Control-Profile**

operations	<b>CSTA-Connection-Cleared-event</b>
	input <a href="#">tns:connectionClearedEvent</a>
	<b>CSTA-Delivered-event</b>
	input <a href="#">tns:deliveredEvent</a>
	<b>CSTA-Diverted-event</b>
	input <a href="#">tns:divertedEvent</a>
	<b>CSTA-Established-event</b>
	input <a href="#">tns:establishedEvent</a>
	<b>CSTA-Failed-event</b>
	input <a href="#">tns:failedEvent</a>
source	<b>CSTA-Held-event</b>
	input <a href="#">tns:heldEvent</a>
	<b>CSTA-Network-Reached-event</b>
	input <a href="#">tns:networkReachedEvent</a>
	<b>CSTA-Retrieved-event</b>
	input <a href="#">tns:retrievedEvent</a>
	<b>CSTA-Service-Initiated-event</b>
	input <a href="#">tns:serviceInitiatedEvent</a>
	<b>CSTA-Transferred-event</b>
	input <a href="#">tns:transferredEvent</a>
	<code>&lt;portType name="CF-CSTA-Conferencing-uaCSTA-Basic-Call-Control-Profile"&gt;</code>
	<code>&lt;operation name="CSTA-Connection-Cleared-event"&gt;</code>
	<code>&lt;output message="tns:connectionClearedEvent"/&gt;</code>
	<code>&lt;/operation&gt;</code>
	<code>&lt;operation name="CSTA-Delivered-event"&gt;</code>
	<code>&lt;input message="tns:deliveredEvent"/&gt;</code>
	<code>&lt;/operation&gt;</code>
	<code>&lt;operation name="CSTA-Diverted-event"&gt;</code>
	<code>&lt;input message="tns:divertedEvent"/&gt;</code>
	<code>&lt;/operation&gt;</code>
	<code>&lt;operation name="CSTA-Established-event"&gt;</code>
	<code>&lt;input message="tns:establishedEvent"/&gt;</code>
	<code>&lt;/operation&gt;</code>

```

<operation name="CSTA-Failed-event">
  <input message="tns:failedEvent"/>
</operation>
<operation name="CSTA-Held-event">
  <input message="tns:heldEvent"/>
</operation>
<operation name="CSTA-Network-Reached-event">
  <input message="tns:networkReachedEvent"/>
</operation>
<operation name="CSTA-Retrieved-event">
  <input message="tns:retrievedEvent"/>
</operation>
<operation name="CSTA-Service-Initiated-event">
  <input message="tns:serviceInitiatedEvent"/>
</operation>
<operation name="CSTA-Transferred-event">
  <input message="tns:transferredEvent"/>
</operation>
</portType>

```

### 34 Conferencing uaCSTA with Advanced Call Control Profile – Port Type Pair

This Clause specifies the CSTA-Conferencing-uaCSTA-Advanced-Call-Control-Profile portType and CF-CSTA-Conferencing-uaCSTA-Advanced-Call-Control-Profile portType. This portType pair specifies the required CSTA WSDL operations to support the CSTA Conferencing uaCSTA Call Control Profile with the Advanced uaCSTA Call Control Profile as specified in ISO/IEC 18051 2.1.3.9.

Implementations may support other operations in addition to the operations specified in this port type pair.

#### porttype **CSTA-Conferencing-uaCSTA-Advanced-Call-Control-Profile**

operations	<b>CSTA-Alternate-Call</b>  input <a href="#"><u>tns:alternateCall</u></a> output <a href="#"><u>tns:alternateCallResponse</u></a> fault <a href="#"><u>tns:negResponse</u></a>  <b>CSTA-Answer-Call</b>  input <a href="#"><u>tns:answerCall</u></a> output <a href="#"><u>tns:answerCallResponse</u></a> fault <a href="#"><u>tns:negResponse</u></a>  <b>CSTA-Clear-Connection</b>  input <a href="#"><u>tns:clearConnection</u></a> output <a href="#"><u>tns:clearConnectionResponse</u></a> fault <a href="#"><u>tns:negResponse</u></a>
------------	--

**CSTA-Conference-Call**

input [tns:conferenceCall](#)  
 output [tns:conferenceCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Consultation-Call**

input [tns:consultationCall](#)  
 output [tns:consultationCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Deflect-Call**

input [tns:deflectCall](#)  
 output [tns:deflectCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Hold-Call**

input [tns:holdCall](#)  
 output [tns:holdCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Make-Call**

input [tns:makeCall](#)  
 output [tns:makeCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Reconnect-Call**

input [tns:reconnectCall](#)  
 output [tns:reconnectCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Retrieve-Call**

input [tns:retrieveCall](#)  
 output [tns:retrieveCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Single-Step-Conference-Call**

input [tns:singleStepConferenceCall](#)  
 output [tns:singleStepConferenceCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Single-Step-Transfer-Call**

input [tns:singleStepTransferCall](#)  
 output [tns:singleStepTransferCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Transfer-Call**

input [tns:transferCall](#)  
 output [tns:transferCallResponse](#)  
 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-Get-Switching-Function-Capabilities**

input [tns:getSwitchingFunctionCaps](#)  
 output [tns:getSwitchingFunctionCapsResponse](#)  
 fault [tns:negResponse](#)

source `<portType name="CSTA-Conferencing-uaCSTA-Advanced-Call-Control-Profile">`  
`<operation name="CSTA-Alternate-Call">`  
`<input message="tns:alternateCall"/>`  
`<output message="tns:alternateCallResponse"/>`  
`<fault name="FaultName" message="tns:negResponse"/>`  
`</operation>`  
`<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-Conference-Call">`  
`<input message="tns:conferenceCall"/>`  
`<output message="tns:conferenceCallResponse"/>`  
`<fault name="FaultName" message="tns:negResponse"/>`  
`</operation>`  
`<operation name="CSTA-Consultation-Call">`  
`<input message="tns:consultationCall"/>`  
`<output message="tns:consultationCallResponse"/>`  
`<fault name="FaultName" message="tns:negResponse"/>`  
`</operation>`

```

<operation name="CSTA-Deflect-Call">
  <input message="tns:deflectCall"/>
  <output message="tns:deflectCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Hold-Call">
  <input message="tns:holdCall"/>
  <output message="tns:holdCallResponse"/>
  <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>

<operation name="CSTA-Reconnect-Call">
  <input message="tns:reconnectCall"/>
  <output message="tns:reconnectCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Retrieve-Call">
  <input message="tns:retrieveCall"/>
  <output message="tns:retrieveCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Single-Step-Conference-Call">
  <input message="tns:singleStepConferenceCall"/>
  <output message="tns:singleStepConferenceCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Single-Step-Transfer-Call">
  <input message="tns:singleStepTransferCall"/>
  <output message="tns:singleStepTransferCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Transfer-Call">
  <input message="tns:transferCall"/>
  <output message="tns:transferCallResponse"/>
  <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-Get-Switching-Function-Capabilities">
  <input message="tns:getSwitchingFunctionCaps"/>
  <output message="tns:getSwitchingFunctionCapsResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
</portType>

```

### porttype **CF-CSTA-Conferencing-uaCSTA-Advanced-Call-Control-Profile**

operations	<b>CSTA-Conferenced-event</b> input <a href="#">tns:conferencedEvent</a>  <b>CSTA-Connection-Cleared-event</b> input <a href="#">tns:connectionClearedEvent</a>  <b>CSTA-Delivered-event</b> input <a href="#">tns:deliveredEvent</a>  <b>CSTA-Diverted-event</b> input <a href="#">tns:divertedEvent</a>  <b>CSTA-Established-event</b> input <a href="#">tns:establishedEvent</a>  <b>CSTA-Failed-event</b> input <a href="#">tns:failedEvent</a>  <b>CSTA-Held-event</b> input <a href="#">tns:heldEvent</a>  <b>CSTA-Network-Reached-event</b> input <a href="#">tns:networkReachedEvent</a>  <b>CSTA-Originated-event</b> input <a href="#">tns:originatedEvent</a>  <b>CSTA-Retrieved-event</b> input <a href="#">tns:retrievedEvent</a>  <b>CSTA-Service-Initiated-event</b> input <a href="#">tns:serviceInitiatedEvent</a>
------------	--

**CSTA-Transferred-event**

```

    input tns:transferredEvent
source <portType name="CF-CSTA-Conferencing-uaCSTA-Advanced-Call-Control-Profile">
  <operation name="CSTA-Conferenced-event">
    <input message="tns:conferencedEvent"/>
  </operation>
  <operation name="CSTA-Connection-Cleared-event">
    <input message="tns:connectionClearedEvent"/>
  </operation>
  <operation name="CSTA-Delivered-event">
    <input message="tns:deliveredEvent"/>
  </operation>
  <operation name="CSTA-Diverted-event">
    <input message="tns:divertedEvent"/>
  </operation>
  <operation name="CSTA-Established-event">
    <input message="tns:establishedEvent"/>
  </operation>
  <operation name="CSTA-Failed-event">
    <input message="tns:failedEvent"/>
  </operation>
  <operation name="CSTA-Held-event">
    <input message="tns:heldEvent"/>
  </operation>
  <operation name="CSTA-Network-Reached-event">
    <input message="tns:networkReachedEvent"/>
  </operation>
  <operation name="CSTA-Originated-event">
    <input message="tns:originatedEvent"/>
  </operation>
  <operation name="CSTA-Retrieved-event">
    <input message="tns:retrievedEvent"/>
  </operation>
  <operation name="CSTA-Service-Initiated-event">
    <input message="tns:serviceInitiatedEvent"/>
  </operation>
  <operation name="CSTA-Transferred-event">
    <input message="tns:transferredEvent"/>
  </operation>
</portType>

```

### 35 Basic uaCSTA Device Feature with Basic Call Control Profile – Port Type Pair

This Clause specifies the CSTA-Basic-uaCSTA-Device-Feature-Basic-Call-Control Profile portType and CF-CSTA-Basic-uaCSTA-Device-Feature-Basic-Call-Control Profile portType. This portType pair specifies the required CSTA WSDL operations to support the CSTA Basic uaCSTA Device Feature Profile with the Basic uaCSTA Call Control Profile as specified in ISO/IEC 18051 2.1.3.10.

Implementations may support other operations in addition to the operations specified in this port type pair.

#### porttype **CSTA-Basic-uaCSTA-Device-Feature-Basic-Call-Control-Profile**

operations	<b>CSTA-Set-Do-Not-Disturb</b>
input	<a href="#"><u>tns:setDoNotDisturb</u></a>
output	<a href="#"><u>tns:setDoNotDisturbResponse</u></a>
fault	<a href="#"><u>tns:negResponse</u></a>
	<b>CSTA-Set-Forwarding</b>
input	<a href="#"><u>tns:setForwarding</u></a>
output	<a href="#"><u>tns:setForwardingResponse</u></a>
fault	<a href="#"><u>tns:negResponse</u></a>
	<b>CSTA-Answer-Call</b>
input	<a href="#"><u>tns:answerCall</u></a>
output	<a href="#"><u>tns:answerCallResponse</u></a>
fault	<a href="#"><u>tns:negResponse</u></a>
	<b>CSTA-Clear-Connection</b>
input	<a href="#"><u>tns:clearConnection</u></a>
output	<a href="#"><u>tns:clearConnectionResponse</u></a>
fault	<a href="#"><u>tns:negResponse</u></a>
	<b>CSTA-Deflect-Call</b>
input	<a href="#"><u>tns:deflectCall</u></a>
output	<a href="#"><u>tns:deflectCallResponse</u></a>
fault	<a href="#"><u>tns:negResponse</u></a>
	<b>CSTA-Hold-Call</b>
input	<a href="#"><u>tns:holdCall</u></a>
output	<a href="#"><u>tns:holdCallResponse</u></a>
fault	<a href="#"><u>tns:negResponse</u></a>
	<b>CSTA-Make-Call</b>
input	<a href="#"><u>tns:makeCall</u></a>
output	<a href="#"><u>tns:makeCallResponse</u></a>
fault	<a href="#"><u>tns:negResponse</u></a>



**CSTA-Retrieve-Call**

input [tns:retrieveCall](#)  
 output [tns:retrieveCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Single-Step-Transfer-Call**

input [tns:singleStepTransferCall](#)  
 output [tns:singleStepTransferCallResponse](#)  
 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](#)

source <portType name="CSTA-Basic-uaCSTA-Device-Feature-Basic-Call-Control-Profile">  
 <operation name="CSTA-Set-Do-Not-Disturb">  
 <input message="tns:setDoNotDisturb"/>  
 <output message="tns:setDoNotDisturbResponse"/>  
 <fault name="FaultName" message="tns:negResponse"/>  
 </operation>  
 <operation name="CSTA-Set-Forwarding">  
 <input message="tns:setForwarding"/>  
 <output message="tns:setForwardingResponse"/>  
 <fault name="FaultName" message="tns:negResponse"/>  
 </operation>  
 <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-Deflect-Call">  
 <input message="tns:deflectCall"/>  
 <output message="tns:deflectCallResponse"/>  
 <fault name="FaultName" message="tns:negResponse"/>  
 </operation>

```

<operation name="CSTA-Hold-Call">
  <input message="tns:holdCall"/>
  <output message="tns:holdCallResponse"/>
  <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>
<operation name="CSTA-Retrieve-Call">
  <input message="tns:retrieveCall"/>
  <output message="tns:retrieveCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Single-Step-Transfer-Call">
  <input message="tns:singleStepTransferCall"/>
  <output message="tns:singleStepTransferCallResponse"/>
  <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>
</portType>

```

#### porttype CF-CSTA-Basic-uaCSTA-Device-Feature-Basic-Call-Control-Profile

operations	<b>CSTA-Do-Not-Disturb-Event</b> input <a href="#">tns:doNotDisturbEvent</a>  <b>CSTA-Forwarding-Event</b> input <a href="#">tns:forwardingEvent</a>  <b>CSTA-Connection-Cleared-event</b> input <a href="#">tns:connectionClearedEvent</a>  <b>CSTA-Delivered-event</b> input <a href="#">tns:deliveredEvent</a>
------------	---

**CSTA-Diverted-event**input [tns:divertedEvent](#)**CSTA-Established-event**input [tns:establishedEvent](#)**CSTA-Failed-event**input [tns:failedEvent](#)**CSTA-Held-event**input [tns:heldEvent](#)**CSTA-Network-Reached-event**input [tns:networkReachedEvent](#)**CSTA-Retrieved-event**input [tns:retrievedEvent](#)**CSTA-Service-Initiated-event**input [tns:serviceInitiatedEvent](#)**CSTA-Transferred-event**input [tns:transferredEvent](#)**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](#)

```

source <portType name="CF-CSTA-Basic-uaCSTA-Device-Feature-Basic-Call-Control-Profile">
  <operation name="CSTA-Do-Not-Disturb-Event">
    <input message="tns:doNotDisturbEvent"/>
  </operation>
  <operation name="CSTA-Forwarding-Event">
    <input message="tns:forwardingEvent"/>
  </operation>
  <operation name="CSTA-Connection-Cleared-event">
    <input message="tns:connectionClearedEvent"/>
  </operation>

```

```
<operation name="CSTA-Delivered-event">
  <input message="tns:deliveredEvent"/>
</operation>
<operation name="CSTA-Diverted-event">
  <input message="tns:divertedEvent"/>
</operation>
<operation name="CSTA-Established-event">
  <input message="tns:establishedEvent"/>
</operation>
<operation name="CSTA-Failed-event">
  <input message="tns:failedEvent"/>
</operation>
<operation name="CSTA-Held-event">
  <input message="tns:heldEvent"/>
</operation>
<operation name="CSTA-Network-Reached-event">
  <input message="tns:networkReachedEvent"/>
</operation>
<operation name="CSTA-Retrieved-event">
  <input message="tns:retrievedEvent"/>
</operation>
<operation name="CSTA-Service-Initiated-event">
  <input message="tns:serviceInitiatedEvent"/>
</operation>
<operation name="CSTA-Transferred-event">
  <input message="tns:transferredEvent"/>
</operation>
</portType>
```

36 Basic uaCSTA Device Feature with Advanced Call Control Profile – Port Type Pair

This Clause specifies the CSTA-Basic-uaCSTA-Device-Feature-Advanced-Call-Control Profile portType and CF-CSTA-Basic-uaCSTA-Device-Feature-Advanced-Call-Control Profile portType. This portType pair specifies the required CSTA WSDL operations to support the CSTA Basic uaCSTA Device Feature Profile with the Advanced uaCSTA Call Control Profile as specified in ISO/IEC 18051 2.1.3.10.

Implementations may support other operations in addition to the operations specified in this port type pair.

porttype **CSTA-Basic-uaCSTA-Device-Feature-Advanced-Call-Control-Profile**

operations	<b>CSTA-Set-Do-Not-Disturb</b>	
	input	<a href="#">tns:setDoNotDisturb</a>
	output	<a href="#">tns:setDoNotDisturbResponse</a>
	fault	<a href="#">tns:negResponse</a>
	<b>CSTA-Set-Forwarding</b>	
	input	<a href="#">tns:setForwarding</a>
	output	<a href="#">tns:setForwardingResponse</a>

fault [tns:negResponse](#)

#### **CSTA-Alternate-Call**

input [tns:alternateCall](#)  
 output [tns:alternateCallResponse](#)  
 fault [tns:negResponse](#)

#### **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-Consultation-Call**

input [tns:consultationCall](#)  
 output [tns:consultationCallResponse](#)  
 fault [tns:negResponse](#)

#### **CSTA-Deflect-Call**

input [tns:deflectCall](#)  
 output [tns:deflectCallResponse](#)  
 fault [tns:negResponse](#)

#### **CSTA-Hold-Call**

input [tns:holdCall](#)  
 output [tns:holdCallResponse](#)  
 fault [tns:negResponse](#)

#### **CSTA-Make-Call**

input [tns:makeCall](#)  
 output [tns:makeCallResponse](#)  
 fault [tns:negResponse](#)

#### **CSTA-Reconnect-Call**

input [tns:reconnectCall](#)  
 output [tns:reconnectCallResponse](#)  
 fault [tns:negResponse](#)

#### **CSTA-Retrieve-Call**

input [tns:retrieveCall](#)  
 output [tns:retrieveCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Single-Step-Transfer-Call**

input [tns:singleStepTransferCall](#)  
 output [tns:singleStepTransferCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Transfer-Call**

input [tns:transferCall](#)  
 output [tns:transferCallResponse](#)  
 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-Get-Switching-Function-Capabilities**

input [tns:getSwitchingFunctionCaps](#)  
 output [tns:getSwitchingFunctionCapsResponse](#)  
 fault [tns:negResponse](#)

source <portType name="CSTA-Basic-uaCSTA-Device-Feature-Advanced-Call-Control-Profile">  
 <operation name="CSTA-Set-Do-Not-Disturb">  
 <input message="tns:setDoNotDisturb"/>  
 <output message="tns:setDoNotDisturbResponse"/>  
 <fault name="FaultName" message="tns:negResponse"/>  
 </operation>  
 <operation name="CSTA-Set-Forwarding">  
 <input message="tns:setForwarding"/>  
 <output message="tns:setForwardingResponse"/>  
 <fault name="FaultName" message="tns:negResponse"/>  
 </operation>  
 <operation name="CSTA-Alternate-Call">  
 <input message="tns:alternateCall"/>  
 <output message="tns:alternateCallResponse"/>  
 <fault name="FaultName" message="tns:negResponse"/>  
 </operation>  
 <operation name="CSTA-Answer-Call">  
 <input message="tns:answerCall"/>  
 <output message="tns:answerCallResponse"/>  
 <fault name="FaultName" message="tns:negResponse"/>  
 </operation>

```

</operation>
<operation name="CSTA-Clear-Connection">
  <input message="tns:clearConnection"/>
  <output message="tns:clearConnectionResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Consultation-Call">
  <input message="tns:consultationCall"/>
  <output message="tns:consultationCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Deflect-Call">
  <input message="tns:deflectCall"/>
  <output message="tns:deflectCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Hold-Call">
  <input message="tns:holdCall"/>
  <output message="tns:holdCallResponse"/>
  <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>
<operation name="CSTA-Reconnect-Call">
  <input message="tns:reconnectCall"/>
  <output message="tns:reconnectCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Retrieve-Call">
  <input message="tns:retrieveCall"/>
  <output message="tns:retrieveCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Single-Step-Transfer-Call">
  <input message="tns:singleStepTransferCall"/>
  <output message="tns:singleStepTransferCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Transfer-Call">
  <input message="tns:transferCall"/>
  <output message="tns:transferCallResponse"/>
  <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-Get-Switching-Function-Capabilities">
  <input message="tns:getSwitchingFunctionCaps"/>
  <output message="tns:getSwitchingFunctionCapsResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
</portType>

```

#### porttype **CF-CSTA-Basic-uaCSTA-Device-Feature-Advanced-Call-Control-Profile**

operations	<b>CSTA-Do-Not-Disturb-Event</b> input <a href="#">tns:doNotDisturbEvent</a>
	<b>CSTA-Forwarding-Event</b> input <a href="#">tns:forwardingEvent</a>
	<b>CSTA-Connection-Cleared-event</b> input <a href="#">tns:connectionClearedEvent</a>
	<b>CSTA-Delivered-event</b> input <a href="#">tns:deliveredEvent</a>
	<b>CSTA-Diverted-event</b> input <a href="#">tns:divertedEvent</a>
	<b>CSTA-Established-event</b> input <a href="#">tns:establishedEvent</a>
	<b>CSTA-Failed-event</b> input <a href="#">tns:failedEvent</a>
	<b>CSTA-Held-event</b> input <a href="#">tns:heldEvent</a>
	<b>CSTA-Network-Reached-event</b>



input [tns:networkReachedEvent](#)

#### CSTA-Originated-event

input [tns:originatedEvent](#)

#### CSTA-Retrieved-event

input [tns:retrievedEvent](#)

#### CSTA-Service-Initiated-event

input [tns:serviceInitiatedEvent](#)

#### CSTA-Transferred-event

input [tns:transferredEvent](#)

source <portType name="CF-CSTA-Basic-uaCSTA-Device-Feature-Advanced-Call-Control-Profile">  
 <operation name="CSTA-Do-Not-Disturb-Event">  
 <input message="tns:doNotDisturbEvent"/>  
 </operation>  
 <operation name="CSTA-Forwarding-Event">  
 <input message="tns:forwardingEvent"/>  
 </operation>  
 <operation name="CSTA-Connection-Cleared-event">  
 <input message="tns:connectionClearedEvent"/>  
 </operation>  
 <operation name="CSTA-Delivered-event">  
 <input message="tns:deliveredEvent"/>  
 </operation>  
 <operation name="CSTA-Diverted-event">  
 <input message="tns:divertedEvent"/>  
 </operation>  
 <operation name="CSTA-Established-event">  
 <input message="tns:establishedEvent"/>  
 </operation>  
 <operation name="CSTA-Failed-event">  
 <input message="tns:failedEvent"/>  
 </operation>  
 <operation name="CSTA-Held-event">  
 <input message="tns:heldEvent"/>  
 </operation>  
 <operation name="CSTA-Network-Reached-event">  
 <input message="tns:networkReachedEvent"/>  
 </operation>  
 <operation name="CSTA-Originated-event">  
 <input message="tns:originatedEvent"/>  
 </operation>  
 <operation name="CSTA-Retrieved-event">

```

        <input message="tns:retrievedEvent"/>
    </operation>
    <operation name="CSTA-Service-Initiated-event">
        <input message="tns:serviceInitiatedEvent"/>
    </operation>
    <operation name="CSTA-Transferred-event">
        <input message="tns:transferredEvent"/>
    </operation>
</portType>

```

### 37 Speaker uaCSTA Device Feature with Basic Call Control Profile – Port Type Pair

This Clause specifies the CSTA-Speaker-uaCSTA-Device-Feature-Basic-Call-Control-Profile portType and CF-CSTA-Speaker-uaCSTA-Device-Feature-Basic-Call-Control-Profile portType. This portType pair specifies the required CSTA WSDL operations to support the CSTA Speaker uaCSTA Device Feature Profile with the Basic uaCSTA Call Control Profile as specified in ISO/IEC 18051 2.1.3.11.

Implementations may support other operations in addition to the operations specified in this port type pair.

#### porttype CSTA-Speaker-uaCSTA-Device-Feature-Basic-Call-Control-Profile

operations	<b>CSTA-Set-Speaker-Mute</b>  input <a href="#">tns:setSpeakerMute</a> output <a href="#">tns:setSpeakerMuteResponse</a> fault <a href="#">tns:negResponse</a>  <b>CSTA-Set-Speaker-Volume</b>  input <a href="#">tns:setSpeakerVolume</a> output <a href="#">tns:setSpeakerVolumeResponse</a> fault <a href="#">tns:negResponse</a>  <b>CSTA-Answer-Call</b>  input <a href="#">tns:answerCall</a> output <a href="#">tns:answerCallResponse</a> fault <a href="#">tns:negResponse</a>  <b>CSTA-Clear-Connection</b>  input <a href="#">tns:clearConnection</a> output <a href="#">tns:clearConnectionResponse</a> fault <a href="#">tns:negResponse</a>  <b>CSTA-Deflect-Call</b>  input <a href="#">tns:deflectCall</a> output <a href="#">tns:deflectCallResponse</a> fault <a href="#">tns:negResponse</a>
------------	---

**CSTA-Hold-Call**

input [tns:holdCall](#)  
 output [tns:holdCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Make-Call**

input [tns:makeCall](#)  
 output [tns:makeCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Retrieve-Call**

input [tns:retrieveCall](#)  
 output [tns:retrieveCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Single-Step-Transfer-Call**

input [tns:singleStepTransferCall](#)  
 output [tns:singleStepTransferCallResponse](#)  
 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](#)

source `<portType name="CSTA-Speaker-uaCSTA-Device-Feature-Basic-Call-Control-Profile">`  
`<operation name="CSTA-Set-Speaker-Mute">`  
`<input message="tns:setSpeakerMute"/>`  
`<output message="tns:setSpeakerMuteResponse"/>`  
`<fault name="FaultName" message="tns:negResponse"/>`  
`</operation>`  
`<operation name="CSTA-Set-Speaker-Volume">`  
`<input message="tns:setSpeakerVolume"/>`  
`<output message="tns:setSpeakerVolumeResponse"/>`  
`<fault name="FaultName" message="tns:negResponse"/>`  
`</operation>`  
`<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-Deflect-Call">
  <input message="tns:deflectCall"/>
  <output message="tns:deflectCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Hold-Call">
  <input message="tns:holdCall"/>
  <output message="tns:holdCallResponse"/>
  <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>
<operation name="CSTA-Retrieve-Call">
  <input message="tns:retrieveCall"/>
  <output message="tns:retrieveCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>
<operation name="CSTA-Single-Step-Transfer-Call">
  <input message="tns:singleStepTransferCall"/>
  <output message="tns:singleStepTransferCallResponse"/>
  <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>
</portType>

```

porttype **CF-CSTA-Speaker-uaCSTA-Device-Feature-Basic-Call-Control-Profile**

operations	<b>CSTA-Speaker-Mute-Event</b>
	input <a href="#">tns:speakerMuteEvent</a>
	<b>CSTA-Speaker-Volume-Event</b>
	input <a href="#">tns:speakerVolumeEvent</a>
	<b>CSTA-Connection-Cleared-event</b>
	input <a href="#">tns:connectionClearedEvent</a>
	<b>CSTA-Delivered-event</b>
	input <a href="#">tns:deliveredEvent</a>
	<b>CSTA-Diverted-event</b>
	input <a href="#">tns:divertedEvent</a>
	<b>CSTA-Established-event</b>
	input <a href="#">tns:establishedEvent</a>
	<b>CSTA-Failed-event</b>
	input <a href="#">tns:failedEvent</a>
source	<b>CSTA-Held-event</b>
	input <a href="#">tns:heldEvent</a>
	<b>CSTA-Network-Reached-event</b>
	input <a href="#">tns:networkReachedEvent</a>
	<b>CSTA-Retrieved-event</b>
	input <a href="#">tns:retrievedEvent</a>
	<b>CSTA-Service-Initiated-event</b>
	input <a href="#">tns:serviceInitiatedEvent</a>
	<b>CSTA-Transferred-event</b>
	input <a href="#">tns:transferredEvent</a>
	<code>&lt;portType name="CF-CSTA-Speaker-uaCSTA-Device-Feature-Basic-Call-Control-Profile"&gt;</code>
	<code>&lt;operation name="CSTA-Speaker-Mute-Event"&gt;</code>
	<code>&lt;input message="tns:speakerMuteEvent"/&gt;</code>
	<code>&lt;/operation&gt;</code>
	<code>&lt;operation name="CSTA-Speaker-Volume-Event"&gt;</code>
	<code>&lt;input message="tns:speakerVolumeEvent"/&gt;</code>
	<code>&lt;/operation&gt;</code>
	<code>&lt;operation name="CSTA-Connection-Cleared-event"&gt;</code>

```
<input message="tns:connectionClearedEvent"/>
</operation>
<operation name="CSTA-Delivered-event">
  <input message="tns:deliveredEvent"/>
</operation>
<operation name="CSTA-Diverted-event">
  <input message="tns:divertedEvent"/>
</operation>
<operation name="CSTA-Established-event">
  <input message="tns:establishedEvent"/>
</operation>
<operation name="CSTA-Failed-event">
  <input message="tns:failedEvent"/>
</operation>
<operation name="CSTA-Held-event">
  <input message="tns:heldEvent"/>
</operation>
<operation name="CSTA-Network-Reached-event">
  <input message="tns:networkReachedEvent"/>
</operation>
<operation name="CSTA-Retrieved-event">
  <input message="tns:retrievedEvent"/>
</operation>
<operation name="CSTA-Service-Initiated-event">
  <input message="tns:serviceInitiatedEvent"/>
</operation>
<operation name="CSTA-Transferred-event">
  <input message="tns:transferredEvent"/>
</operation>
</portType>
```

38 Speaker uaCSTA Device Feature with Advanced Call Control Profile – Port Type Pair

This Clause specifies the CSTA-Speaker-uaCSTA-Device-Feature-Advanced-Call-Control-Profile portType and CF-CSTA-Speaker-uaCSTA-Device-Feature-Advanced-Call-Control-Profile portType. This portType pair specifies the required CSTA WSDL operations to support the CSTA Speaker uaCSTA Device Feature Profile with the Advanced uaCSTA Call Control Profile as specified in ISO/IEC 18051 2.1.3.11.

Implementations may support other operations in addition to the operations specified in this port type pair.

porttype **CSTA-Speaker-uaCSTA-Device-Feature-Advanced-Call-Control-Profile**

operations	<b>CSTA-Set-Speaker-Mute</b>
input	<a href="#">tns:setSpeakerMute</a>
output	<a href="#">tns:setSpeakerMuteResponse</a>
fault	<a href="#">tns:negResponse</a>

**CSTA-Set-Speaker-Volume**

input [tns:setSpeakerVolume](#)  
 output [tns:setSpeakerVolumeResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Alternate-Call**

input [tns:alternateCall](#)  
 output [tns:alternateCallResponse](#)  
 fault [tns:negResponse](#)

**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-Consultation-Call**

input [tns:consultationCall](#)  
 output [tns:consultationCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Deflect-Call**

input [tns:deflectCall](#)  
 output [tns:deflectCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Hold-Call**

input [tns:holdCall](#)  
 output [tns:holdCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Make-Call**

input [tns:makeCall](#)  
 output [tns:makeCallResponse](#)  
 fault [tns:negResponse](#)

**CSTA-Reconnect-Call**

input [tns:reconnectCall](#)  
 output [tns:reconnectCallResponse](#)  
 fault [tns:negResponse](#)

#### CSTA-Retrieve-Call

input [tns:retrieveCall](#)  
 output [tns:retrieveCallResponse](#)  
 fault [tns:negResponse](#)

#### CSTA-Single-Step-Transfer-Call

input [tns:singleStepTransferCall](#)  
 output [tns:singleStepTransferCallResponse](#)  
 fault [tns:negResponse](#)

#### CSTA-Transfer-Call

input [tns:transferCall](#)  
 output [tns:transferCallResponse](#)  
 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-Get-Switching-Function-Capabilities

input [tns:getSwitchingFunctionCaps](#)  
 output [tns:getSwitchingFunctionCapsResponse](#)  
 fault [tns:negResponse](#)

source `<portType name="CSTA-Speaker-uaCSTA-Device-Feature-Advanced-Call-Control-Profile">`  
`<operation name="CSTA-Set-Speaker-Mute">`  
`<input message="tns:setSpeakerMute"/>`  
`<output message="tns:setSpeakerMuteResponse"/>`  
`<fault name="FaultName" message="tns:negResponse"/>`  
`</operation>`  
`<operation name="CSTA-Set-Speaker-Volume">`  
`<input message="tns:setSpeakerVolume"/>`  
`<output message="tns:setSpeakerVolumeResponse"/>`  
`<fault name="FaultName" message="tns:negResponse"/>`  
`</operation>`  
`<operation name="CSTA-Alternate-Call">`  
`<input message="tns:alternateCall"/>`  
`<output message="tns:alternateCallResponse"/>`  
`<fault name="FaultName" message="tns:negResponse"/>`  
`</operation>`



```

<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-Consultation-Call">
  <input message="tns:consultationCall"/>
  <output message="tns:consultationCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Deflect-Call">
  <input message="tns:deflectCall"/>
  <output message="tns:deflectCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Hold-Call">
  <input message="tns:holdCall"/>
  <output message="tns:holdCallResponse"/>
  <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>

<operation name="CSTA-Reconnect-Call">
  <input message="tns:reconnectCall"/>
  <output message="tns:reconnectCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Retrieve-Call">
  <input message="tns:retrieveCall"/>
  <output message="tns:retrieveCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Single-Step-Transfer-Call">
  <input message="tns:singleStepTransferCall"/>
  <output message="tns:singleStepTransferCallResponse"/>
  <fault name="FaultName" message="tns:negResponse"/>
</operation>

<operation name="CSTA-Transfer-Call">
  <input message="tns:transferCall"/>
  <output message="tns:transferCallResponse"/>

```

```

    <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-Get-Switching-Function-Capabilities">
    <input message="tns:getSwitchingFunctionCaps"/>
    <output message="tns:getSwitchingFunctionCapsResponse"/>
    <fault name="FaultName" message="tns:negResponse"/>
  </operation>
</portType>

```

### porttype CF-CSTA-Speaker-uaCSTA-Device-Feature-Advanced-Call-Control-Profile

operations	<b>CSTA-Speaker-Mute-Event</b> input <a href="#"><u>tns:speakerMuteEvent</u></a>
	<b>CSTA-Speaker-Volume-Event</b> input <a href="#"><u>tns:speakerVolumeEvent</u></a>
	<b>CSTA-Connection-Cleared-event</b> input <a href="#"><u>tns:connectionClearedEvent</u></a>
	<b>CSTA-Delivered-event</b> input <a href="#"><u>tns:deliveredEvent</u></a>
	<b>CSTA-Diverted-event</b> input <a href="#"><u>tns:divertedEvent</u></a>
	<b>CSTA-Established-event</b> input <a href="#"><u>tns:establishedEvent</u></a>
	<b>CSTA-Failed-event</b> input <a href="#"><u>tns:failedEvent</u></a>
	<b>CSTA-Held-event</b> input <a href="#"><u>tns:heldEvent</u></a>
	<b>CSTA-Network-Reached-event</b> input <a href="#"><u>tns:networkReachedEvent</u></a>
	<b>CSTA-Originated-event</b> input <a href="#"><u>tns:originatedEvent</u></a>
	<b>CSTA-Retrieved-event</b> input <a href="#"><u>tns:retrievedEvent</u></a>

**CSTA-Service-Initiated-event**input [tns:serviceInitiatedEvent](#)**CSTA-Transferred-event**input [tns:transferredEvent](#)

```

source <portType name="CF-CSTA-Speaker-uaCSTA-Device-Feature-Advanced-Call-Control-Profile">
  <operation name="CSTA-Speaker-Mute-Event">
    <input message="tns:speakerMuteEvent"/>
  </operation>
  <operation name="CSTA-Speaker-Volume-Event">
    <input message="tns:speakerVolumeEvent"/>
  </operation>
  <operation name="CSTA-Connection-Cleared-event">
    <input message="tns:connectionClearedEvent"/>
  </operation>
  <operation name="CSTA-Delivered-event">
    <input message="tns:deliveredEvent"/>
  </operation>
  <operation name="CSTA-Diverted-event">
    <input message="tns:divertedEvent"/>
  </operation>
  <operation name="CSTA-Established-event">
    <input message="tns:establishedEvent"/>
  </operation>
  <operation name="CSTA-Failed-event">
    <input message="tns:failedEvent"/>
  </operation>
  <operation name="CSTA-Held-event">
    <input message="tns:heldEvent"/>
  </operation>
  <operation name="CSTA-Network-Reached-event">
    <input message="tns:networkReachedEvent"/>
  </operation>
  <operation name="CSTA-Originated-event">
    <input message="tns:originatedEvent"/>
  </operation>
  <operation name="CSTA-Retrieved-event">
    <input message="tns:retrievedEvent"/>
  </operation>
  <operation name="CSTA-Service-Initiated-event">
    <input message="tns:serviceInitiatedEvent"/>
  </operation>
  <operation name="CSTA-Transferred-event">
    <input message="tns:transferredEvent"/>
  </operation>
</portType>

```

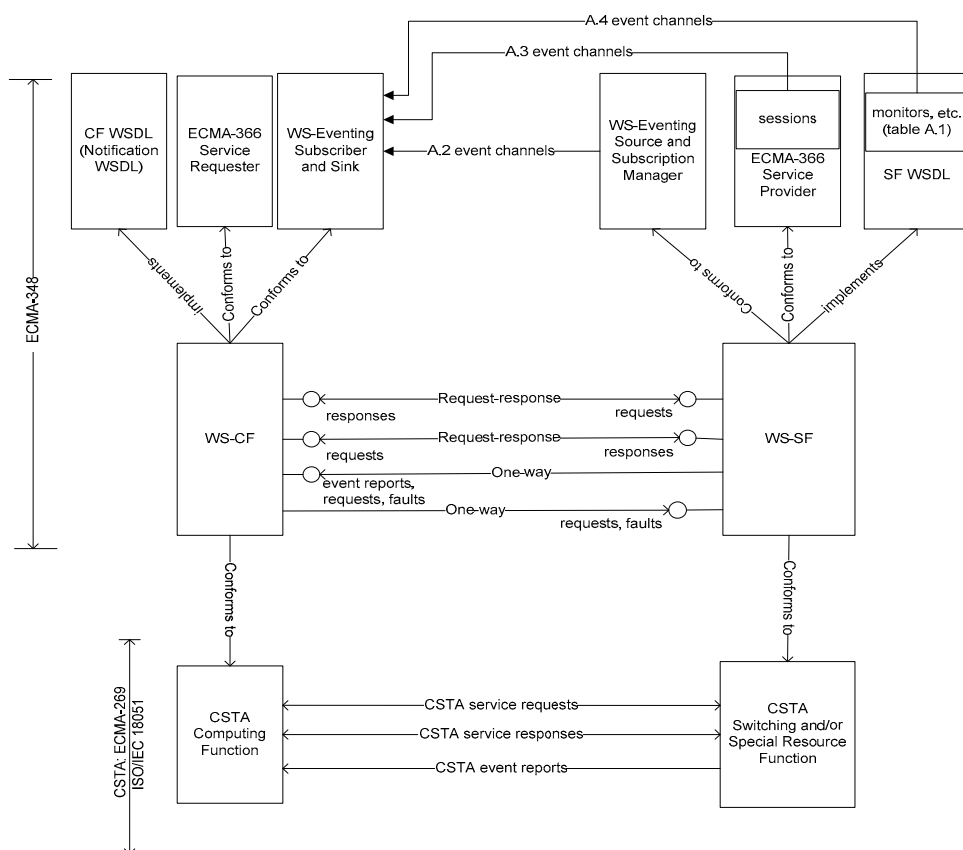
## Annex A (normative)

### Event Channel Establishment

#### A.1 General

An event channel represents a relation between an event source that generates event notifications and an event sink that receives the event notifications. This Annex specifies mechanisms to establish event channels between a WS-SF and a WS-CF using WS-Eventing and mechanisms to establish ISO/IEC 18051 Implicit Associations based on the event channels.

In particular, this Annex treats a WS-SF as an event source and a WS-CF as an event sink, and all the operations in the Computing Function WSDL as event notifications from the WS-SF. A WS-SF may also contain other event sources, such as ISO/IEC 25437 sessions, CSTA monitors and registrations, that generate subsets of the notifications specified in the Computing Function WSDL. To support these event sources, this Annex specifies three types of event channels in A.2, A.3 and A.4 as illustrated in the following diagram.



A WS-SF shall implement the WS-Eventing Event Source, Subscription Manager, wse:NotifyTo element and the extensions defined in this Annex in conformance to WS-Eventing. It shall publish its Computing Function WSDL as a Notification WSDL specified in A.2 of WS-Eventing. It shall implement at least one of the event channels specified in A.2 and A.3 and may implement event channels specified in A.4. The Switching Function shall send all notifications in the Computing Function WSDL as unwrapped. It may also send the ISO/IEC 18051 event reports as wrapped.

A WS-CF shall implement the WS-Eventing Event Sink, Subscriber, wse:NotifyTo and the extensions defined in this Annex in conformance to WS-Eventing. It shall implement the operations in the Computing Function WSDL for the notifications it intends to receive. It shall establish event channels using the mechanisms supported by the WS-SF. It shall implement the unwrapped format and may implement the wrapped format defined by WS-Eventing.

To establish an event channel, a WS-CF shall send a WS-Eventing Subscribe message to a WS-SF as specified in the following clauses. Once the WS-CF receives a positive response, the event channel is established.

Once an event channel is established, all event notifications from its source shall be sent over the channel as specified in the following clauses.

To terminate an event channel, the WS-CF shall send a WS-Eventing Unsubscribe message to the WS-SF as specified by WS-Eventing.

## A.2 Event Channel from Switching Function

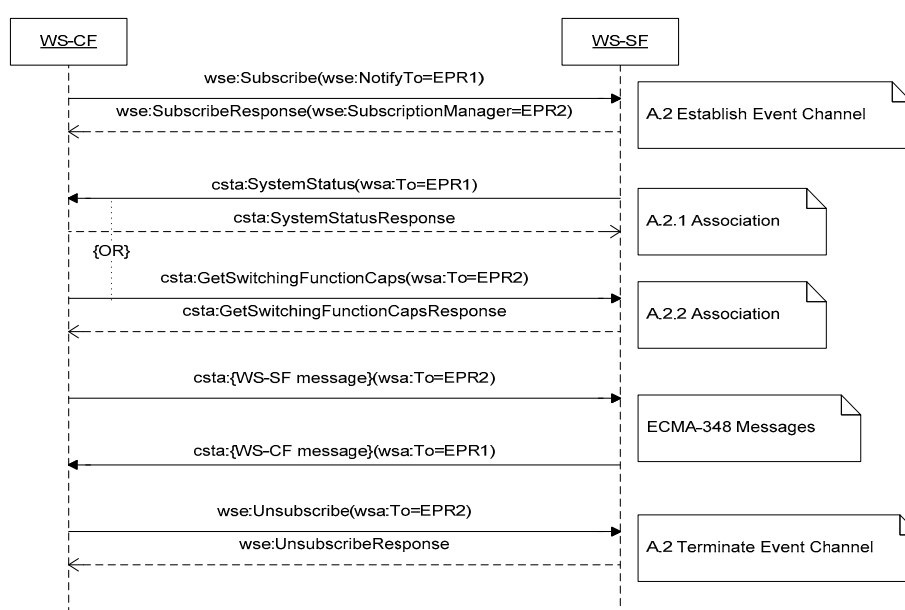
The event source of this type of event channels is a WS-SF. To establish an event channel of this type, a WS-CF shall send a WS-Eventing Subscribe message with a valid wse:NotifyTo element to the WS-SF endpoint. On success, the WS-SF shall respond with a WS-Eventing SubscribeResponse message with a valid wse:SubscriptionManager element.

Once an event channel is established, the WS-CF and WS-SF shall establish an implicit association over the event channel as specified in A.2.1 or A.2.2.

To send a message over an established association, the WS-CF shall address the message to the wse:SubscriptionManager Endpoint Reference (EPR), and the WS-SF shall address the message to the wse:NotifyTo Endpoint Reference, as specified by WS-Addressing.

To terminate an event channel and the corresponding association, a WS-CF shall send a WS-Eventing Unsubscribe message.

The message exchange is illustrated in the following UML sequence diagram.



### A.2.1 ISO/IEC 18051 7.1 Implicit Association

A WS-SF shall send a System Status message addressed to the wse:NotifyTo Endpoint Reference in the established event channel. Once the WS-SF receives a positive response, the association is established.

### A.2.2 ISO/IEC 18051 7.2 Implicit Association

A WS-CF shall send a Get Switching Function Capabilities message addressed to the wse:SubscriptionManager Endpoint Reference in the established event channel. Once the WS-CF receives a positive response, the association is established.

## A.3 Event Channel from Session

The event source of this type of event channels is an application session in ISO/IEC 25437. To establish an event channel of this type, a WS-CF shall use the mechanism specified in Clause 7 and Annex A of ISO/IEC 25437.

## A.4 Event Channel from Event Sources in Switching Function (Option)

The event sources of this type of event channels include monitors, registrations and other objects maintained by a WS-SF. These event sources are contained within an association, which itself may be an event source with associated event channels as specified in A.2 and A.3.

Table A.1 lists these event sources with the corresponding ISO/IEC 18051 clauses. The first column of the table lists the ISO/IEC 18056 elements that identify the event sources. The second column lists the WS-SF services that create and remove the event sources. The third column lists the event notifications associated with each event source.

**Table A.1 — Event Sources of Switching Function**

Event source identification	Event source management	Event Notifications
sysStatRegisterID	14.1.2 System Register 14.1.4 System Register Cancel	14.1.3 System Register Abort 14.2.1 Request System Status 14.2.2 System Status 14.2.3 Switching Function Capabilities Changed 14.2.4 Switching Function Devices Changed
monitorCrossRefID	15.1.2 Monitor Start 15.1.3 Monitor Stop	15.1.3 Monitor Stop 17.2.1-18 Call Control Events 18.2.1-5 Call Associated Events 19.2.1-2 Media Attachment Events 21.2.1-11 Physical Device Events 22.2.1-14 Logical Device Events 23.1.1-4 Device Maintenance Events 26.2.1-18 Voice Services Events 28.3.1 Private Event

**Table A.1 — Event Sources of Switching Function (concluded)**

routeRegisterReqID	20.1.1 Route Register 20.1.3 Route Register Cancel	20.1.2 Route Register Abort 20.2.1 Re-Route 20.2.2 Route End 20.2.3 Route Reject Negative Acknowledgement 20.2.4 Route Request 20.2.5 Route Select Negative Acknowledgement 20.2.6 Route Used
ioRegisterReqID	24.1.1 I/O Register 24.1.3 I/O Register Cancel	24.1.2 I/O Register Abort 24.2.1 Data Path Resumed 24.2.2 Data Path Suspended 24.2.3 Fast Data 24.2.6 Send Data 24.2.8 Start Data Path 24.2.9 Stop Data Path 24.2.10 Suspend Data Path
dCollCrossRefID	25.1.5 Start Data Collection 25.1.6 Stop Data Collection	25.1.1 Data Collected 25.1.2 Data Collection Resumed 25.1.3 Data Collection Suspended 25.1.6 Stop Data Collection 25.1.7 Suspend Data Collection
cdrCrossRefID	27.1.4 Start Call Detail Records Transmission 27.1.5 Stop Call Detail Records Transmission	27.1.1 Call Detail Records Notification 27.1.2 Call Detail Records Report 27.1.5 Stop Call Detail Records Transmission
locCrossRefID	28.1.7 Start Location Tracking Session 28.1.8 Stop Location Tracking Session	28.1.3 Location Tracking Session Resumed 28.1.4 Location Tracking Session Suspended 28.1.6 Location Information Report
escapeReigsterID	29.1.1 Escape Register 29.1.3 Escape Register Cancel	29.1.2 Escape Register Abort 29.2.1 Escape 29.3.1 Private Event

To establish an event channel of this type over an association, a WS-CF shall send a WS-Eventing Subscribe message with a valid wse:NotifyTo element and an extension element [event source] that identifies an event source as listed in Table A.1:

```

<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:csta="http://www.ecma-international.org/standards/ecma-323/csta/ed4">
  <S:Header>
    [event source]
    <wsa:Action>http://www.w3.org/2011/03/ws-evt/Subscribe</wsa:Action>
  </S:Header>
  <S:Body>...</S:Body>
</S:Envelope>

```

For example, the following message attempts to establish an event channel from a monitor [event source] within an application session:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:wse="http://www.w3.org/2011/03/ws-evt"
  xmlns:aps="http://www.ecma-international.org/standards/ecma-354/appl_session" >
  <S:Header>
    <aps:sessionID>5555</aps:sessionID>
    <csta:monitorCrossRefID>6666</csta:monitorCrossRefID>
    <wsa:Action>http://www.w3.org/2011/03/ws-evt/Subscribe</wsa:Action>
  </S:Header>
  <S:Body>
    <wse:Subscribe>
      <wse:Delivery><wse:NotifyTo>
        <wsa:Address>http://www.example.com/CSTA_sink</wsa:Address>
      </wse:NotifyTo></wse:Delivery>
    </wse:Subscribe>
  </S:Body>

```

Whenever an event channel is established from an event source within an association, the event channels associated with the association, if any, shall be disabled until all the event channels for the event source are terminated. This constraint avoids duplicated event notifications from an event source over different event channels.

Once an event source is removed from a WS-SF, all event channels from it shall be terminated.

If this option is not supported, a WS-SF shall return a SOAP 1.1 fault message with these properties:

- [faultcode]="ExplicitSubscriptionNotSupported"
- [faultstring]="Explict subscription to event source [event source] is not supported"

If an event source does not exist, a WS-SF shall return a SOAP 1.1 fault message with these properties:

- [faultcode]="UnknownEventSource"
- [faultstring]="The event source [event source] is invalid"



## Annex B (informative)

### Example SOAP 1.1/HTTP Binding

The following is an example of a WSDL description with SOAP/HTTP binding. In order to keep the example simple, only one CSTA operation (Answer Call) is included.

```
<definitions>

  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:csta="http://www.ecma-international.org/standards/ecma-323/csta/ed6"
  xmlns:tns="http://www.ecma-international.org/standards/ecma-348/csta-wsdl/ed5"
  targetNamespace="http://www.ecma-international.org/standards/ecma-348/csta-wsdl/ed5">

  <!--WSDL Type definitions see 6.2-->

  <types>
    <xs:schema targetNamespace="http://www.ecma-international.org/standards/ecma-348/csta-wsdl/ed5">
      <xs:import namespace="http://www.ecma-international.org/standards/ecma-323/csta/ed6"
        schemaLocation="csta.xsd"/>
    </xs:schema>
  </types>

  <!--WSDL Message definitions see 6.3-->

  <message name="answerCall">
    <part name="parameter" element="csta:AnswerCall"/>
  </message>
  <message name="answerCallResponse">
    <part name="parameter" element="csta:AnswerCallResponse"/>
  </message>
  <message name="negResponse">
    <part name="parameter" element="csta:CSTACodeError"/>
  </message>

  <!--WSDL Operation and PortType definitions see 6.4 and 6.5-->

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

  <!--WSDL Binding definitions see 6.6-->

  <binding name="SOAP_HTTP" type="tns:samplePortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="CSTA-Answer-Call">
      <soap:operation soapAction="http://CSTA_web_service.com/answerCall" style="document"/>
      <input>
```

```

        <soap:body use="literal"/>
    </input>
    <output>
        <soap:body use="literal"/>
    </output>
    <fault name="FaultName"/>
</fault>
</operation>
</binding>

<!--WSDL Port and Service definitions see 6.7 and 6.8-->

<service name="CSTA_SOAP_Service">
    <port name="CSTA_Sample-Port" binding="tns:SOAP_HTTP">
        <soap:address location="http://CSTA_web_service.com"/>
    </port>
</service>
</definitions>

```

The fault message specified in 6.3.3 is bound to the detail subelement of SOAP 1.1 Fault as specified in the following template:

```

<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:csta="http://www.ecma-international.org/standards/ecma-323/csta/ed6">
  <S:Body>
    <S:Fault>
      <detail><csta:ErrorCode>...</csta:CSTAErroCode></detail>
    </S:Fault>
  </S:Body>
</S:Envelope>

```

The other subelements of SOAP 1.1 Fault: faultcode, faultstring and faultactor, are not constrained.

## **Annex C** (informative)

### **WSDL Considerations**

WSDL version 1.1 has several limitations:

- Does not define fault messages with one-way and Notification operation types
- Does not define concrete SOAP/HTTP bindings for Notification and solicit-response operation types
- A specific operation can only exist in one port type.

To address these limitations in WSDL 1.1, this International Standard:

1. Specifies faults associated with CSTA services without positive acknowledgement as one-way operation.
2. Specifies Computing Function WSDL, which contains only request-response and one-way operations and can have concrete bindings, to represent operations initiated by the Switching Function.
3. Specifies the port types by CSTA services each containing distinct WSDL operations.

