TECHNICAL REPORT

ISO/IEC TR 22767

First edition
2005-08-15

Information technology — Telecommunications and information exchange between systems — Using CSTA for SIP phone user agents (uaCSTA)

Technologies de l'information — Télécommunications et échange d'information entre systèmes — Utilisation de CSTA pour agents d'utilisateurs de téléphone de SIP (uaCSTA)

Reference number
ISO/IEC TR 22767:2005(E)

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

Published in Switzerland

Contents

Page

Foreword

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.

In exceptional circumstances, the joint technical committee may propose the publication of a Technical Report of one of the following types:

— type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;

— type 3, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

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 TR 22767, which is a Technical Report of type [3], was prepared by Ecma International (as ECMA TR/87) and was adopted, under a special "fast-track procedure", by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommitte SC 6, *Telecommunications and information exchange between systems*, in parallel with its approval by national bodies of ISO and IEC.

Introduction

This Technical Report illustrates how CSTA can be used over a SIP session to control and observe SIP user agents (uaCSTA).

This Technical Report is part of a suite of Ecma CSTA Phase III Standards and Technical Reports.

All of the Standards and Technical Reports in this suite are based upon the practical experience of Ecma member companies and each one represents a pragmatic and widely based consensus.

Information technology — Telecommunications and information exchange between systems — Using CSTA for SIP phone user agents (uaCSTA)

1 Scope

The Session Initiation Protocol (SIP) is a control (signalling) protocol for creating, modifying, and terminating sessions with one or more participants. These sessions include Internet telephone calls, multimedia distribution, and multimedia conferences.

CSTA standardizes a very powerful and flexible set of application services to observe and control voice and non-voice media calls as well as control and observe non-call related features.

This Technical Report describes how CSTA can be used to provide a subset of CSTA call control functionality, called 1st party call control, for SIP user agents. The term uaCSTA (for user agent CSTA) refers to transporting ECMA-323 (CSTA XML) messages over a SIP session.

uaCSTA leverages SIP mechanisms to provide a highly featured, robust, and extensible set of features to support applications in the Enterprise environment.

uaCSTA can be implemented by several different types of SIP user agents:

- directly by a SIP user agent on a SIP phone
- uaCSTA can also be implemented by a SIP B2BUA to augment 3PCC functionality
- by a proxy server that is front-ending a PBX.

2 Purpose

- Describe the relevant portions of the CSTA Standards – The two primary CSTA Standards, Services for Computer Supported Telecommunications Applications (ECMA-269) and the XML Protocol for CSTA (ECMA-323), are relatively large standards (combined over 1100 pages). Due to their size it is sometimes difficult for SIP developers without prior knowledge of CSTA standards to quickly find the relevant parts of the CSTA standards needed to implement basic features. This TR shows the relevant CSTA concepts and how they can be used to implement a CSTA-based application protocol without having to read all of the CSTA Standards.

- Terminology – Although many of the concepts are similar, different terms are used in SIP and CSTA to describe the same concepts. Since CSTA is designed to be protocol independent, it is helpful to show how the abstract terminology of CSTA is mapped to SIP specific terminology.

- Extensibility - A SIP phone that is being developed to support a specific application may initially need to only support a small subset of the features standardized in CSTA. As the types and complexity of applications using these devices increase, it is expected that these devices will need to support additional, more advanced, features standardized by CSTA, similar to features supported by other types of phones in Enterprise environments. This TR shows how basic features can be extended to support a rich standards-based feature set for applications.

- Interoperability - In order to encourage interoperability of applications and phones supporting this application protocol, additional CSTA Profiles, which include minimal sets of CSTA functionality, are described. These profiles can be extended by implementations to provide a more complete set of call and devices features commonly used by Enterprise applications.

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.

This TR provides informative examples of how to use CSTA concepts as an application protocol for SIP user agents. The following Ecma Standards should be used as the definitive references for CSTA.

ECMA-269, *Services for Computer Supported Telecommunications Applications (CSTA) Phase III, 6th edition, June 2004 (International Standard ISO/IEC 18051)*

ECMA-323, *XML Protocol for Computer Supported Telecommunications Applications (CSTA) Phase III, 3rd edition, June 2004 (International Standard ISO/IEC 18056)*

The following IETF references provide information on the SIP features referenced in this TR:

RFC 3261, *SIP: Session Initiation Protocol, Rosenberg, J. et al, IETF, June 2002*

4 Terminology

4.1 General terms

The following table summarizes some of the commonly used terminologies used in this Technical Report.

application	Refers to all components that control and observe uaCSTA instances that may be present in an application/computing domain such as an application residing on any type of computing device (desktop PC, mobile computing device, CTI middleware, SIP B2BUA, SIP UA, etc.).
uaCSTA	user agent CSTA – This refers the mechanism to transport CSTA messages over a SIP session, and the type of call control, called 1st party call control, that allows an application to control and observe only the device or devices it is directly involved with. A desktop application controlling its associated phone, for example.
uaCSTA device	Refers to a SIP UA, such as a SIP or PBX phone, that supports the CSTA features over SIP as described in this TR.
phone	A SIP UA that supports calls. It is also a PBX device that supports calls. Examples of the types of phones that can be supported by this application protocol are described in clause 6.
B2BUA	A back-to-back-user-agent is a type of user agent that acts both as a user agent server (receives requests) and acts as a user agent client (generates requests). B2BUA is used to implement SIP third party call control (3PCC) features. uaCSTA could be used to augment the existing B2BUA functionality for enhanced call control.

4.2 SIP/CSTA Terminology Mappings

The following table shows some of the common SIP terms and how they are referenced using CSTA and vice versa.

SIP Term	CSTA Term	Notes
User Agent or UA	device	A SIP UA can be modelled as a type of CSTA device. Whenever the term device is used in this TR, it is referring to a SIP UA.
SIP URI	device identifier or deviceID	A SIP URI can be used to address a SIP UA or a PBX phone in the same way that a CSTA device identifier can be used to address a CSTA device. One of the formats of a Device Identifier is URI. Whenever the term DeviceID is used in this TR, it is referring to a URI format of CSTA deviceID. Refer to Annex A for a description of the ways to represent a user and a user's device with a URI.
TEL URI	device identifier or deviceID	A TEL URI can be used to address a device or a dialled number in the same way that a CSTA device identifier can be used to address a CSTA device. Refer to Annex A for a description of the ways to represent a device and a dialled number with a TEL URI.

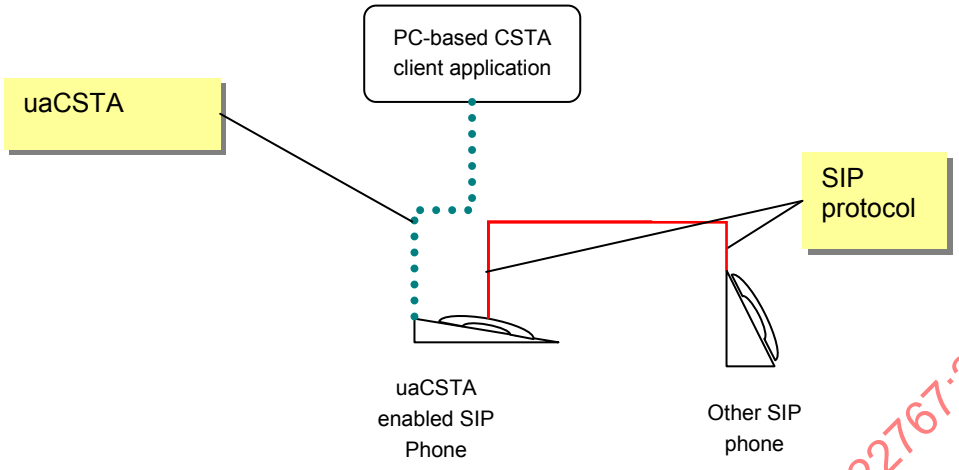
5 Example Environments for uaCSTA

The following figures illustrate how uaCSTA is used to pass ECMA-323 messages between an application and a user agent that supports this protocol. uaCSTA provides a standards-based protocol for applications to use to request a UA to invoke features.

The term uaCSTA device refers to a SIP UA, such as a SIP phone, that supports the ECMA-323 messages over SIP as described in this Technical Report.

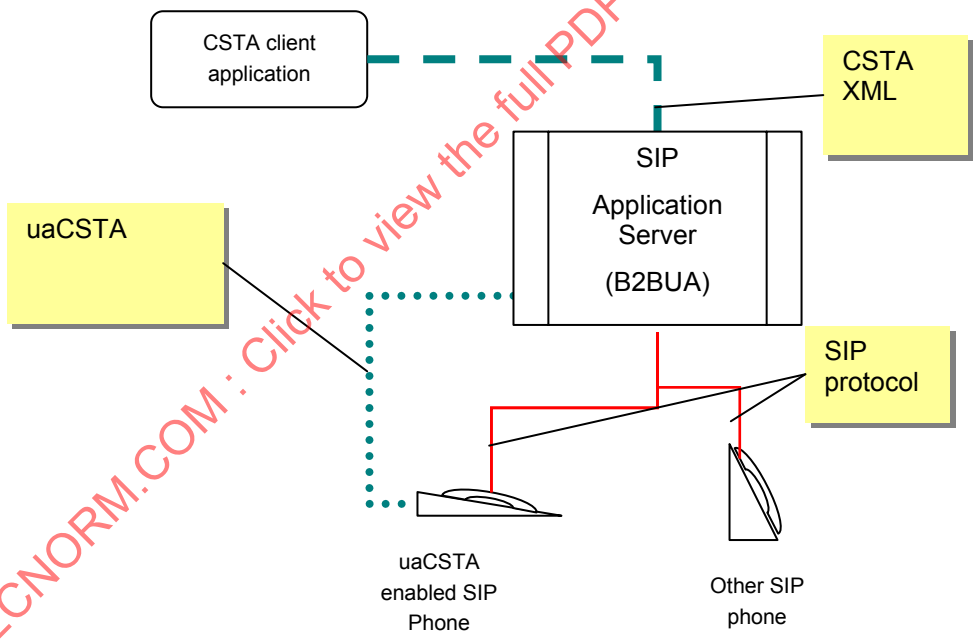
5.1 Controlling and Observing a SIP Phone

In this environment, a PC-based application can use uaCSTA to directly control and observe its associated SIP UA phone.



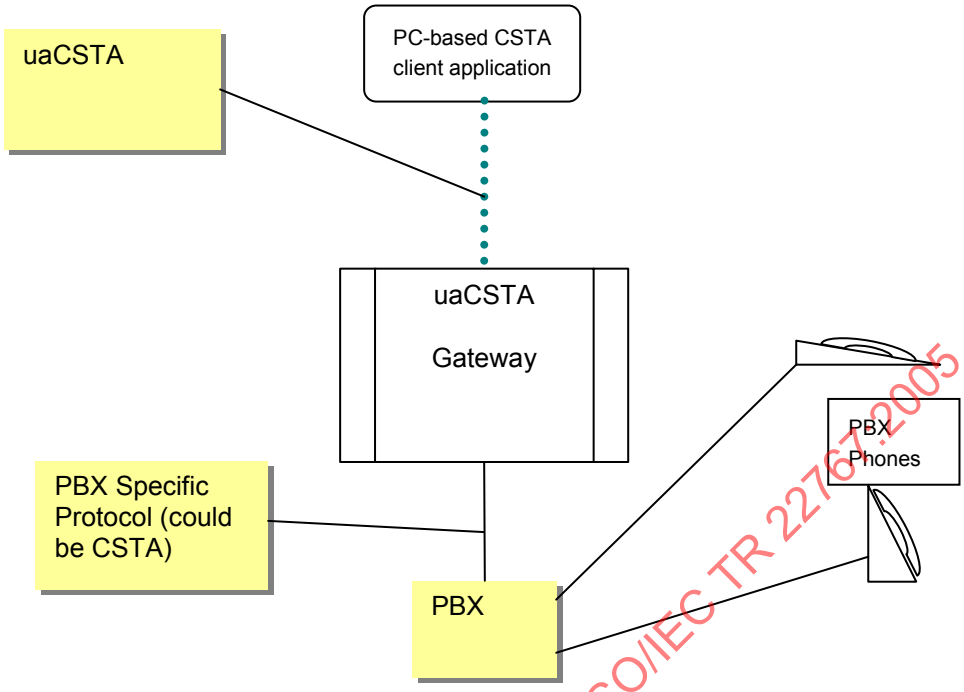
5.2 Controlling and Observing a SIP Phone by Augmenting B2BUA Functionality

In this environment, uaCSTA is used to augment existing B2BUA functionality to invoke features on the SIP phone not possible with standard SIP.



5.3 Controlling a PBX Phone

In this environment, a PC-based application can use uaCSTA to control its associated PBX phone that is part of a PBX/IP Switch. A SIP/CSTA Gateway is a type SIP UA that terminates SIP and converts ECMA-323 messages to/from an application to/from PBX dependent protocols. The uaCSTA Gateway allows an application to control only its associated device(s) – (1st party call control).



6 Example User Agent Configurations

CSTA defines two elements that are relevant for user agents:

- A logical element – the set of attributes/features/services that have an association with the control and/or observation of calls at a UA. For example (line) appearances and attributes such as forwarding. Some phones provide more than one addressable logical elements (e.g. multi line phones). Some phones share a logical element among multiple physical devices (e.g. bridged appearance phones).

- A physical element – the set of attributes/features/services that have any association with the control or observation of the physical components (speaker, microphone, display, etc.) of a UA.

Some UAs provide a single address which can be used to address all of its (logical and physical) elements while other UAs provide multiple addresses so that requests can be directed to a specific logical element on the UA (e.g. a specific line on a multi-line phone).

This chapter discusses the typical types of phones that can be supported by this application protocol, the types of elements supported by phone type, and how applications can address each element.

6.1 Single Line Phone UA

A single line phone is a type of user agent that supports only one logical element (addressable line) not shared with any other device and a single physical element (see 6.1.3.3.2 of ECMA-269). The single line can support one or more calls at the same time.

For a single line phone, a single CSTA deviceID (e.g. SIP URI) is used to reference both the phone's logical and physical elements.

For example if the phone has registered with the contact address of sip:1234@domain1, the various elements of the phone could be addressed via CSTA services as follows:

- phone's logical element with a call control feature: To originate a call from the single line, an application would provide sip:1234@domain1 as the callingDevice in the CSTA Make Call service.

- phone's logical element with a logical device feature: To control the forwarding feature for the single line, an application would provide sip:1234@domain1 as the deviceID in the CSTA Set Forwarding service.

- phone's physical element with a physical device feature: To set the speaker mute for the device, an application would provide sip:1234@domain1 as the deviceID in the CSTA Set Speaker Mute service.

6.2 Multi Line Phone UA

A multi line phone is a type of user agent that supports more than one addressable line (see 6.1.3.3.3 of ECMA-269). Each line can support one or more calls at the same time.

For a multi line phone, there are multiple logical elements (lines) and a single physical element.

For a multi line phone, each logical element (line) is addressed with a unique CSTA deviceID (e.g. SIP URI). CSTA services that reference a phone's logical element use the deviceID (e.g. SIP contact URI) associated with the desired line. CSTA services that apply to the physical element address it via any one of the SIP contact URIs since they all, in this phone configuration, refer to the same device.

For example if the phone has registered with the contacts of sip:1234@domain1 and sip:5678@domain1, the various elements of the phone could be addressed via CSTA services as follows:

- phone's logical component with a call control feature:

  — To originate a call from the line addressed via sip:1234@domain1, an application would provide sip:1234@domain1 as the callingDevice in the CSTA Make Call service.

  — To originate a call from the line addressed via sip:5678@domain1, an application would provide sip:5678@domain1 as the callingDevice in the CSTA Make Call service.

- phone's logical element with a logical device feature:

  — To control the forwarding feature for the line addressed via sip:1234@domain1, an application would provide sip:1234@domain1 as the deviceID in the CSTA Set Forwarding service.

  — To control the forwarding feature for the line addressed via sip:5678@domain1, an application would provide sip:5678@domain1 as the deviceID in the CSTA Set Forwarding service.

- phone's physical element with a physical device feature: To set the speaker mute for the device, an application would provide either one of the following:

  — sip:1234@domain1 as the deviceID in the CSTA Set Speaker Mute service. Since this contact is only associated with one physical device, it would be unambiguous.

  — sip:5678@domain1 as the deviceID in the CSTA Set Speaker Mute service. Since this contact is only associated with one physical device, it would be unambiguous.

6.3 Bridged Appearance Phone and other Advanced UA Configurations

A bridged-appearance phone is a type of user agent that contains a logical element (e.g. a line) shared with another phone (see 6.1.3.3.5 of ECMA-269).

CSTA specifies many different kinds of bridged appearances that reflect the different behaviours and different capabilities for calls at shared appearances. (See bridged calls in section of ECMA-269.)

uaCSTA does not specify (nor preclude) the behaviour or the addressing of bridged appearances and other advanced UA configurations.

7 SIP Transport Mechanism for CSTA Messages

This clause describes how:

- SIP is used to establish a CSTA application session
- CSTA service request and response messages are transported over SIP
- a CSTA Monitor is started and CSTA events are transported over SIP.

The mechanism described here uses the SIP INVITE and INFO methods to transport ECMA-323 messages provided in a SIP message body.

The SIP Content-type used for an ECMA-323 message is application/csta+xml. This MIME media type is used specifically for encapsulating ECMA-323 messages.

A user agent using this mechanism could be any type of SIP UA such as a SIP phone or a SIP user agent that is front-ending a PBX, like the CSTA/SIP server as shown in Clause 5.

NOTE The SIP mechanism described here to establish a CSTA application session and transport ECMA-323 messages is one of many possible mechanisms. Other options such as using SIP SUBSCRIBE/NOTIFY methods with a CSTA event package(s) could be used. SIP extensions to methods such as REFER may provide another option.

7.1 Establishing a CSTA Application Session

Before ECMA-323 service requests are sent to a SIP UA, a CSTA application session must be established between the application and the SIP UA.

ECMA-269 specifies a number of options for establishing a CSTA application session. This mechanism uses the “Implicit Association created using CSTA Request System Status” option as described in 7.2 of ECMA-269.

The application creates an application session by establishing a SIP dialog with the user agent using a SIP INVITE method that includes a Content-Disposition header indicating “signal” and "handling=required" to mandate support for the application/csta+xml MIME type.

An ECMA-323 Request System Status service request is included in the SIP INVITE body with the Content-Type application/csta+xml.

The Request-URI header in the SIP INVITE could specify a particular user agent instance (e.g. globally unique contact URI) or a SIP Address-of-Record URI (a user’s published address). If the Request-URI specifies a SIP Address-of-Record for which multiple devices have registered contacts, the application should be aware that a SIP proxy may pick a specific contact to forward this INVITE to or the proxy may parallel fork the INVITE to all of the contacts. This behaviour depends upon the proxy design, policy, scripting, etc. In general if the INVITE is forked, the application will get the contact URIs from all instances and can choose which specific contact it wishes to monitor and/or control using uaCSTA.

If the UA supports this MIME type and can establish this application session, it responds to this SIP Invite with a 200 OK (line 2 below) that includes an ECMA-323 Request System Status service response in the message body. The application sends a SIP ACK to complete the dialog establishment (line 3).

If the UA does not support this MIME type, the SIP UA must provide a 415 (Unsupported Media Type) response as defined in IETF RFC 3261.

The application-control session exists for as long as the application is interested in controlling the SIP user agent which might, for some deployments, persist as long as the user agent is operational. Once the application-control session is no longer needed, it can be terminated using the SIP BYE method.

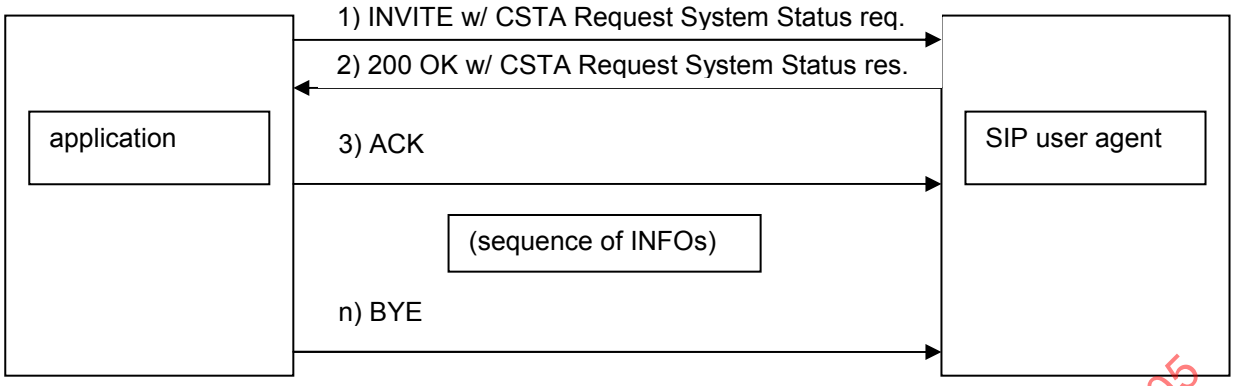


Figure 1 — Establishing a CSTA Application Session using the SIP INVITE method

7.2 Transporting CSTA Service Requests and Responses

Once a SIP dialog has been created for the CSTA application session, ECMA-323 service requests can be sent using the SIP INFO method.

An ECMA-323 service request is provided in the body of a SIP INFO method. The SIP INFO method is sent using the SIP dialog created above.

An ECMA-323 (positive or negative) service response is provided in the body of a 200 OK method.

The CSTA response can be correlated to the CSTA request using the SIP CSEQ headers associated with the SIP dialog used for the application control session.

ECMA-323 service requests can be sent by either the application or the UA. Steps 1 and 2 in the figure below show how the SIP INFO method is used for an application initiated request using the CSTA Make Call service as an example. Steps 3 and 4 show a UA initiated request using the CSTA System Status service as an example.

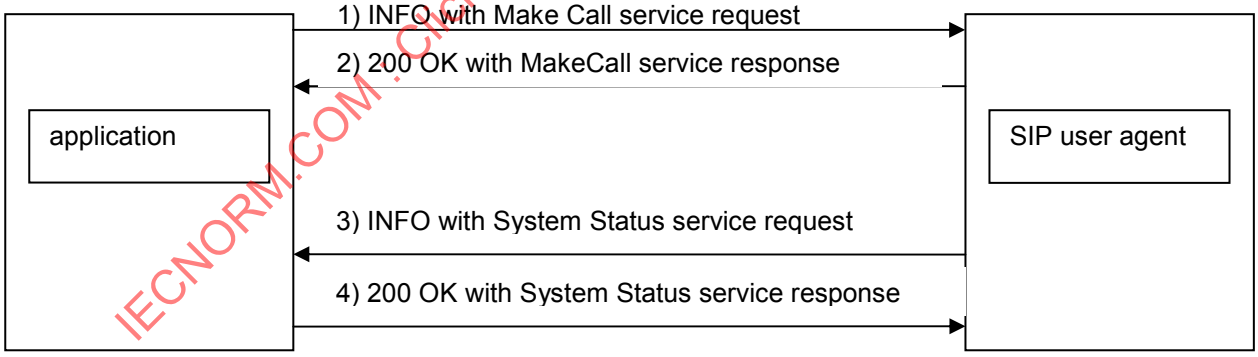


Figure 2 — Sending CSTA Service Requests and Service Responses using the SIP INFO method

7.3 Starting a Monitor and Transporting CSTA Events

An application needs to start a CSTA monitor before it can observe changes to CSTA calls and features at a UA via CSTA events. To start a monitor, an ECMA-323 Monitor Start service request is sent in the body of a SIP INFO method (line 1 below).

The user agent responds to the SIP INFO with a SIP 200 OK (line 2 below). The SIP UA provides an ECMA-323 Monitor Start response in the SIP 200 OK message.

The SIP UA generates subsequent ECMA-323 events in a SIP INFO method (line 3 below).

A supported event is generated whenever a CSTA connection changes state for a call that the UA is involved with or when changes occur in the state of a physical or logical feature, as per the monitoring requirements for supported CSTA services and events as specified in ECMA-269.

ECMA-323 events are generated regardless of what caused the event, i.e. either through application initiated features or manually invoked features.

When an application wants to terminate an existing monitor, it generates a SIP INFO method with an ECMA-323 Monitor Stop service request (line 5 below).

If the SIP UA terminates the monitor the SIP UA generates a SIP INFO method with an ECMA-323 Monitor Stop service request.

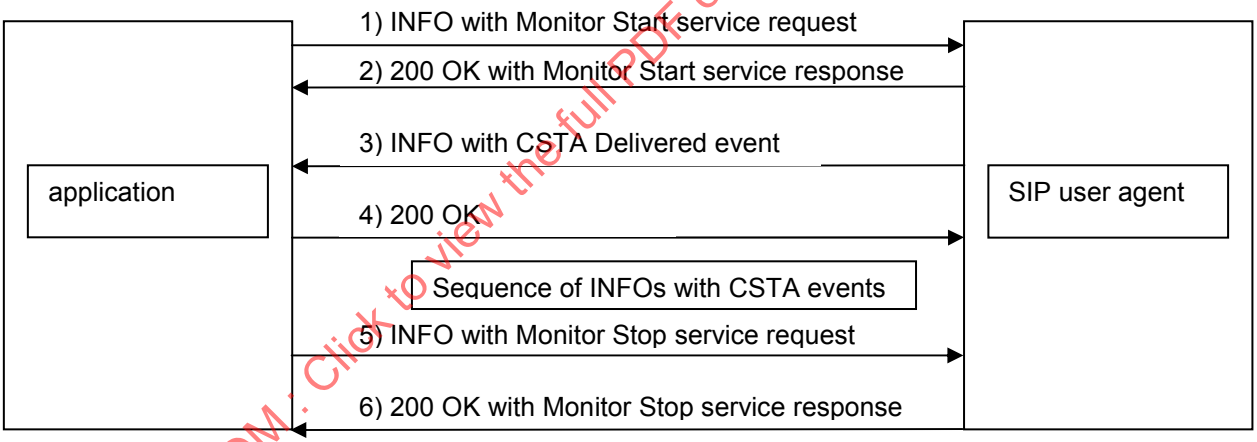


Figure 3 — Establishing a CSTA Monitor and sending CSTA events using the SIP INFO method

8 uaCSTA Profiles

Since many CSTA features are optional, and to enhance application portability across different CSTA implementations, CSTA standards require a minimal subset of functionality as conformance criteria.

ECMA-269 specifies this minimal subset of functionality via a set of standardized profiles. In order to claim conformance to CSTA at least one profile must be supported. The following profiles, specified in ECMA-269 6th Edition, most closely match the call control and device control services and events needed by an application that is managing a SIP user agent. Note that multiple profiles may be supported.

- Minimal uaCSTA Call Control Profile – provides support for making a call, answering an incoming call, deflecting a call, clearing a call, transferring a call in a single step, holding a call, and retrieving a call.

- Basic uaCSTA Call Control Profile – in addition to the features in the Minimal uaCSTA Call Control Profile, this profile includes the ability to monitor call activity for supported CSTA call and device services at a UA.

- Advanced uaCSTA Call Control Profile – in addition to the features in the Basic uaCSTA Call Control Profile, this profile includes the ability to start a consultation call, reconnect a call, alternate between two calls, and transfer a call (two step transfer).

- Conferencing uaCSTA Call Control Profile – in addition to the features in the Basic or the Advanced uaCSTA Call Control Profile, this profile includes the ability to conference another device into an existing call at the user agent in two steps (Conference Call) or to conference another device into a call in one step (Single Step Conference Call).

- Basic uaCSTA Device Feature Profile – in addition to the features in the Basic or the Advanced uaCSTA Call Control Profile, this profile provides support for setting the forwarding feature and setting the Do Not Disturb feature at a UA.

- Speaker uaCSTA Device Feature Profile – in addition to the features in the Basic or the Advanced uaCSTA Call Control Profile, this profile provides support for setting the speaker mute feature and setting the speaker volume feature at a UA.

8.1 Minimal uaCSTA Call Control Profile

8.1.1 Services

The following CSTA services are included in the *Minimal uaCSTA Call Control Profile:*

- Answer call – Answers an alerting call at a UA.

- Clear Connection – Clears a connection at a UA.

- Deflect Call – Moves a connection away from the deflecting UA. The deflecting UA is no longer involved with the call after the Deflect Call service is completed.

- Hold Call – Holds a call at a holding UA.

- Make Call – Makes a call from an originating UA.

- Retrieve Call – Retrieves a call at a retrieving UA.

- Single Step Transfer – Transfers a call to another device in a single step. A transferring UA is no longer involved with the call after the Single Step Transfer service is completed.

8.1.2 Events

There are no CSTA events specified as part of this profile. This profile assumes that an application uses mechanisms outside of this Standard (SIP Subscribe/Notify, for example) to obtain call/connection information that can be used in CSTA services rather than using CSTA events to obtain this information.

8.2 Basic uaCSTA Call Control Profile

8.2.1 Services

The following CSTA services are included in the *Basic uaCSTA Call Control Profile*:

- Answer call – Answers an alerting call at a UA.

- Clear Connection – Clears a connection at a UA.

- Deflect Call – Moves a connection away from a deflecting UA. A deflecting UA is no longer involved with the call after the Deflect Call service is completed.

- Hold Call – Holds a call at a holding UA.
- Make Call – Makes a call from an originating UA.
- Retrieve Call – Retrieves a call at a retrieving UA.
- Single Step Transfer – Transfers a call to another device in a single step. A transferring UA is no longer involved with the call after the Single Step Transfer service is completed.
- Monitor Start – Establishes a device-type monitor on a UA.
- Monitor Stop – Terminates an existing monitor.

8.2.2 Events

The following CSTA events are included in the *uaCSTA Call Control Profile:*

- Connection Cleared – Indicates that a UA has disconnected from a call.
- Delivered – Indicates that a call is alerting a UA.
- Diverted – Indicates that a UA has redirected a call to another device and is no longer involved with the call.
- Established – Indicates that a UA has answered or has been connected to a call.
- Failed – Indicates that a call cannot be completed (e.g. call has encountered a busy device).
- Held – Indicates that a call at a UA is on hold.
- Network Reached – For an outbound call, indicates that a call has been connected to an external network via a Network Interface Device.
- Retrieved – Indicates that a call at a UA has been retrieved.
- Service Initiated – Indicates that a call is prompting the originating UA to go off hook (part of a CSTA Make Call service) or the UA is requesting service.
- Transferred – Indicates that a call at the UA has been transferred to another location.

8.3 Advanced uaCSTA Call Control Profile

8.3.1 Services

The following CSTA services are included in the *Advanced uaCSTA Call Control Profile*:

- Alternate Call – Places an existing call on hold and retrieves a previously held call at a UA.
- Answer Call – Answers an alerting call at a UA.
- Clear Connection – Clears a connection at a UA.
- Consultation Call – Places an existing call on hold at the UA and initiates a new call from the UA.
- Deflect Call – Moves a connection away from the deflecting UA. The deflecting UA is no longer involved with the call after the Deflect Call service is completed.

- Hold Call – Holds a call at the holding UA.
- Make Call – Makes a call from an originating UA.
- Reconnect Call – Clears an existing connected call and retrieves a call on hold at a UA.
- Retrieve Call – Retrieves a call at a retrieving UA.
- Single Step Transfer – Transfers a connected call to another device without placing the call on hold. The transferring UA is no longer involved with the call after this service is completed.
- Transfer Call – Merges two calls at the UA into one call. As a result of the Transfer, the device is no longer involved with the call.
- Monitor Start – Establishes a device-type monitor on a UA.
- Monitor Stop – Terminates an existing monitor.

8.3.2 Events

The following CSTA events are included in the *Advanced uaCSTA Call Control Profile:*

- Originated – For an outbound call, indicates that an originating UA is connected to the call.
- Connection Cleared – Indicates that a UA has disconnected from a call.
- Delivered – Indicates that a call is alerting a UA.
- Diverted – Indicates that a UA has redirected a call to another device and is no longer involved with the call.
- Established – Indicates that an UA has answered or has been connected to a call.
- Failed – Indicates that a call cannot be completed (e.g. call has encountered a busy device).
- Held – Indicates that a call at the UA is on hold.
- Network Reached – For an outbound call, indicates that the call has been connected to an external network via a Network Interface Device.
- Retrieved – Indicates that a call at the UA has been retrieved.
- Service Initiated – Indicates that a call is prompting the originating UA to go off hook (as part of a CSTA Make Call service) or the UA is requesting service.
- Transferred – Indicates that a call at a UA has been transferred to another location.

8.4 Conferencing uaCSTA Call Control Feature Profile

8.4.1 Services

The CSTA services in the *Conferencing uaCSTA Call Control Profile* must include the services in either the Basic or the Advanced uaCSTA Call Control Profile plus:

- Conference Call – Conferences two calls together at a UA.
- Single Step Conference Call – Adds another device to an existing call at a UA in one step.

8.4.2 Events

The CSTA events in the *Conferencing uaCSTA Call Control Profile* must include the events in either the Basic or the Advanced uaCSTA Call Control Profile plus:

- Conferenced – Indicates that a conference call has been created at a UA.

8.5 Basic uaCSTA Device Feature Profile

8.5.1 Services

The CSTA services in the *Basic uaCSTA Device Feature Profile* must include the services in either the Basic or the Advanced uaCSTA Call Control Profile plus:

- Set Do Not Disturb – Sets the DND feature status at a UA. This is a CSTA logical device feature (see Clause 12).
- Set Forwarding – Sets the forwarding feature status at a UA. This is a CSTA logical device feature (see Clause 12).

8.5.2 Events

The CSTA events in the *Basic uaCSTA Device Feature Profile* must include the events in either the Basic or the Advanced uaCSTA Call Control Profile plus:

- Do Not Disturb – Indicates that the DND feature status has changed at a UA.
- Forwarding – Indicates that the forwarding feature status has changed at a UA.

8.6 Speaker uaCSTA Device Feature Profile

8.6.1 Services

The CSTA services in the *Speaker uaCSTA Device Feature Profile* must include the services in either the Basic or the Advanced uaCSTA Call Control Profile plus:

- Set Speaker Mute – Sets the speaker mute feature status at a UA. This is a CSTA physical device feature (see Clause 11).
- Set Speaker Volume – Sets the speaker volume feature status at a UA. This is a CSTA physical device feature (see Clause 11).

8.6.2 Events

The CSTA events in the *Speaker uaCSTA Device Feature Profile* must include the events in either the Basic or the Advanced uaCSTA Call Control Profile plus:

- Speaker Mute – Indicates that the speaker mute feature status has changed at a UA.
- Speaker Volume – Indicates that the speaker volume feature status has changed at a UA.

9 CSTA Calls and Connections

This Clause summarizes the fundamental concepts regarding CSTA calls and connections.

Most CSTA call control services are applied to a CSTA connection. A CSTA connection is referenced via a CSTA connection identifier. A CSTA connection identifier consists of a call identifier and a device (UA) identifier.

In a typical 1st party call control environment, an application manipulates only the CSTA device (or devices) and CSTA connections directly associated (via provisioning, for example) with the application. A device identifier is included in a CSTA connection identifier to allow the application to reference any device associated with the application. For example, a phone may support multiple deviceIDs (one per addressable line appearance).

9.1 CSTA Connection State Model

A CSTA application is informed of connection state transitions (via ECMA-323 call control events) by placing a monitor on a device via an associated address (e.g. this is how an application "listens" for incoming calls).

Each CSTA connection in a call is associated with a connection state. CSTA specifies a connection state model (see ECMA-269, Figure 6-19) that consists of the following connection states:

- Alerting –Indicates an incoming call at a device. Typically, the connection may be ringing or it may be in a pre-alerting (e.g. offered) condition.
- Connected – Indicates that a connection is actively participating in a call. This connection state can be the result of an incoming or outgoing call.
- Failed – Indicates that call progression has stalled. Typically, this could indicate that an outgoing call attempt that encountered a busy device.
- Held – Indicates that a device is no longer actively participating in a call. For implementations that support multiple calls per a single device (i.e. line), a connection could be Held while the line is used to place another call (consultation transfer on a single line, for example).
- Initiated – A transient state, usually indicating that the device is initiating a service (e.g. dial tone) or is being prompted to go off hook.
- Null – There is no relationship between the call and the device.
- Queued – Indicates that the call is temporarily suspended at a device (e.g. call has been parked, camped on).

The CSTA Connection State is provided in ECMA-323 events.

9.2 Connection State Transitions for CSTA Calls

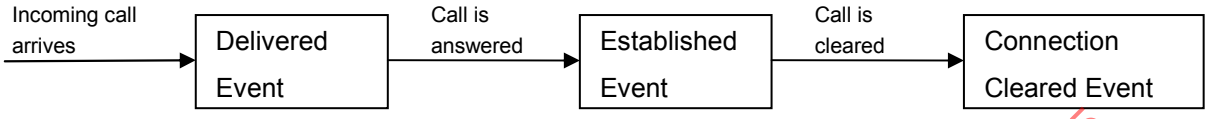
9.2.1 Incoming Call

The following figure illustrates the CSTA events for an incoming call to a UA. The connection state of the UA (called connection) is indicated in parenthesis.

- Offered Event (Alerting) – Indicates a call is in a pre-alerting state. This gives an application an opportunity to influence the routing of the call before it actually rings the device. The application can either accept, clear, or deflect the call to another UA. If the application does not send any service within an (implementation dependent) time, the device will be alerted normally. If the Offered event is not supported by the UA or requested by an application, the call will continue directly to an alerting state.
- Delivered Event (Alerting) – Indicates a call is alerting. Calls that are "auto-answered" do not send this event. A CSTA Answer Call service can be used to answer the call. This results in an Established event.

- Established Event (Connected) – Indicates a call has been answered. Media path has been established. The CSTA Clear Connection service can be used to clear the call. A Connection Cleared event is generated as the result of the Clear Connection service.

- Connection Cleared Event (Null) - Indicates a connection has cleared. This can be the result of the Clear Connection service or as the result of any party clearing from the call.



9.2.2 Outgoing Call

The following figure illustrates the CSTA events for an outgoing call from a UA. The connection state of the UA (originating connection) is indicated in parenthesis. This sequence could be the result of a CSTA Make Call service.

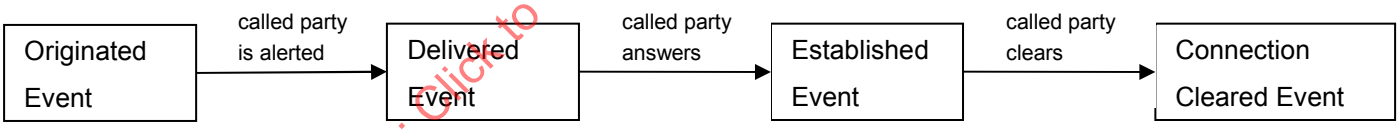
- Service Initiated Event (Initiated) – Indicates that the originating connection at a UA is prompting the user to go off hook. This event is not generated if the originating UA is not prompted (i.e., auto-originated).

- Originated Event (Connected) – Indicates that the originating connection at a UA is connected.

- Delivered Event (Connected) – Indicates the call is alerting the called party.

- Established Event (Connected) – Indicates the called party has answered the call. The media path has been established.

- Connection Cleared Event (Null) – Indicates a connection at a UA has cleared.



10 Call Control

This Clause shows examples of how CSTA is used to control calls at a SIP UA.

Note that, in addition to these features included in this Clause, there are many other call control features specified in ECMA-269.

10.1 Alternate Call

An application uses this service to place an existing call on hold and then retrieves a previously held call. This can also be used to place an existing call on hold and then connect to an alerting or queued call at the same UA.

The UA should provide a positive response if it can successfully alternate the call.

The UA should provide a negative response with an appropriate error code if it cannot successful perform the service. Some examples:

- The specified calls at the UA are not in the appropriate states (e.g. two held calls).

- The UA does not support this service.

10.1.1 Service Request

The common Service Request elements are:

- heldCall – this mandatory element contains the connection identifier of the held connection. The connection identifier consists of the logical address of the device (e.g. line) and the call identifier of the call. The application uses the contents of the connection element provided by either CSTA events, or by other mechanisms (e.g. SIP event packages).

- activeCall – this mandatory element contains the connection identifier of the active connection. The connection identifier consists of the logical address of the device (e.g. line) and the call identifier of the call. The application uses the contents of the connection element provided by either CSTA events, or by other mechanisms (e.g. SIP event packages).

In this example, the application indicates that the held and active calls at UA sip:tom1@domain.com should be alternated. As a result of this service the active call becomes a held call and the held call becomes an active call.

```
<?xml version="1.0" encoding="UTF-8"?>
<AlternateCall xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
 <heldCall>
  <callID>123456789</callID>
  <deviceID>sip:tom1@domain.com</deviceID>
 </heldCall>
 <activeCall>
  <callID>5678</callID>
  <deviceID>sip:tom1@domain.com</deviceID>
 </activeCall>
</AlternateCall>
```

10.1.2 Positive Service Response

There are no common Service Response elements in the positive service response.

This example indicates that the UA has alternated the active and held calls.

```
<?xml version="1.0" encoding="UTF-8"?>
<AlternateCallResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"/>
```

10.1.3 Negative Service Response

The CSTAErrorCode message is used to indicate a negative response.

The common elements are:

- A mandatory error category element. There is a choice of seven error categories each with a set of error values. See ECMA-323 clause 9.19 for a list of all possible error codes. Some commonly used error codes are:

  — operation (serviceNotSupported) – UA does not support the alternate call service

  — operation (invalidConnectionIdentifier) – one of the connection identifiers in the request is not valid

— stateIncompatibility (invalidConnectionState) – one (or both) of the calls is not in the appropriate state.

This example shows a negative response because the Alternate Call service is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>serviceNotSupported</operation>
</CSTAErrorCode>
```

This example shows a negative response because one of the connection identifiers in the request is invalid.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>invalidConnectionIdentifier</operation>
</CSTAErrorCode>
```

This example shows a negative response because of a privilege violation.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>privilegeViolation</operation>
</CSTAErrorCode>
```

10.2 Answer Call

An application uses this service to answer an alerting call at a UA.

The UA should provide a positive response if it can successfully answer the call.

The UA should provide a negative response with an appropriate error code if it cannot successful answer the call. Some examples:

- The UA cannot uniquely identify an alerting call based upon the connection identifier in the request.
- The specified call at the UA is not in the alerting state.
- The UA does not support this service.

10.2.1 Service Request

The common Service Request elements are:

- callToBeAnswered – this mandatory element contains the connection identifier of the alerting connection. The connection identifier consists of the logical address of the device (e.g. line) and the call identifier of the call. The application uses the contents of the connection element provided by either CSTA events, or by other mechanisms (e.g. SIP event packages).

In this example, the application indicates that the alerting call with the SIP call-id of 123456789, remote-tag of 1222, and a local-tag of 7777 at the UA sip:tom@domain.com should be answered. (The SIP call-id, local-tag, and remote-tag were obtained from a SIP event package.)

```
<?xml version="1.0" encoding="UTF-8"?>
<AnswerCall xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
```

```
    <callToBeAnswered>
        <callID>call-id:123456789, remote-tag=1222, local-tag=7777</callID>
        <deviceID>sip:tom@domain.com</deviceID>
    </callToBeAnswered>
</AnswerCall>
```

10.2.2 Positive Service Response

There are no common Service Response elements in the positive service response.

This example indicates that the UA is answering the requested call.

```
<?xml version="1.0" encoding="UTF-8"?>
<AnswerCallResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"/>
```

10.2.3 Negative Service Response

The CSTAErrorCode message is used to indicate a negative response.

The common elements are:

- A mandatory error category element. There is a choice of seven error categories each with a set of error values. See ECMA-323 clause 9.19 for a list of all possible error codes. Some commonly used error codes are:

  — operation (serviceNotSupported) – UA does not support the answer call service

  — operation (invalidConnectionIdentifier) – the connection identifier in the request is not valid

  — stateIncompatibility (invalidConnectionState) – call is not in the alerting state.

This example shows a negative response because there is no alerting call at the device.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
    <stateIncompatibility>invalidConnectionState</stateIncompatibility>
</CSTAErrorCode>
```

This example shows a negative response because the Answer Call service is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
    <operation>serviceNotSupported</operation>
</CSTAErrorCode>
```

This example shows a negative response because the connection identifier in the request is invalid.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
    <operation>invalidConnectionIdentifier</operation>
</CSTAErrorCode>
```

This example shows a negative response because of a privilege violation.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>privilegeViolation</operation>
</CSTAErrorCode>
```

10.3 Clear Connection

An application uses this service to clear an existing call at a UA.

The UA should provide a positive response if it can successfully clear the call.

The UA should provide a negative response with an appropriate error code if it cannot successfully clear the call. Some examples:

- The UA cannot uniquely identify the call based upon the connection identifier in the request.
- The UA does not support this service.
- The call has already been cleared at the UA.

10.3.1 Service Request

The common Service Request elements are:

- connectionToBeCleared – this mandatory element contains the connection identifier of the call to be cleared. The connection identifier consists of the logical address of the device (e.g. line) and the call identifier of the call. The application uses the contents of the connection element provided by either CSTA events, or by other mechanisms (e.g. SIP event packages).

In this example, the application indicates that call with the call identifier of 123456789 at the UA sip:tom1@domain.com should be cleared.

```
<?xml version="1.0" encoding="UTF-8"?>
<ClearConnection xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <connectionToBeCleared>
      <callID>123456789</callID>
      <deviceID>sip:tom1@domain.com</deviceID>
   </connectionToBeCleared>
</ClearConnection>
```


10.3.2 Positive Service Response

There are no common Service Response elements in the positive service response.

This example indicates that the UA is clearing the requested call.

```
<?xml version="1.0" encoding="UTF-8"?>
<ClearConnectionResponse xmlns="http://www.ecma-international.org/standards/ecma-
323/csta/ed3"/>
```

10.3.3 Negative Service Response

The CSTAErrorCode message is used to indicate a negative response.

The common elements are:

- A mandatory error category element. There is a choice of seven error categories each with a set of error values. See ECMA-323 clause 9.19 for a list of all possible error codes. Some commonly used error codes are:

  — operation (serviceNotSupported) – UA does not support the service

  — operation (invalidConnectionIdentifier) – the connection identifier in the request is not valid

  — operation (noConnectionToClear) – there is no call at the UA associated with the connection identifier in the request.

This example shows a negative response because the Clear Connection Call service is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>serviceNotSupported</operation>
</CSTAErrorCode>
```

This example shows a negative response because the connection identifier in the request is invalid.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>invalidConnectionIdentifier</operation>
</CSTAErrorCode>
```

This example shows a negative response because there is no connection at the UA to clear.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <stateIncompatibility>invalidConnectionState</stateIncompatibility>
</CSTAErrorCode>
```

10.4 Consultation Call

An application uses the Consultation Call service to place a call on hold and to originate a new call from the same device.

10.4.1 Service Request

The common Service Request elements are:

- existingCall – this mandatory element specifies the existing connection
- consultedDevice – this mandatory element specifies the consulted device.

In this example, the application wants to place an existing call on hold and place another call from the same UA to the number 14085551212.

```
<?xml version="1.0" encoding="UTF-8"?>
<ConsultationCall xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <existingCall>
      <callID>123456789</callID>
      <deviceID>sip:tom1@domain.com</deviceID>
   </existingCall>
   <consultedDevice>14085551212</consultedDevice>
```

```
</ConsultationCall>
```

10.4.2 Positive Service Response

This example shows the positive response to the Consultation Call request.

The common Service Response elements are:

- initiatedCall connection identifier – this element provides the connection identifier that is used to reference the consulted connection. The connection identifier consists of the logical address of the device (e.g. line) and the call identifier of the call.

In this example, the UA indicates the originating connectionID for the call at the UA.

```
<?xml version="1.0" encoding="UTF-8"?>
<ConsultationCallResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <initiatedCall>
      <callID>55555</callID>
      <deviceID>sip:tom1@domain.com</deviceID>
   </initiatedCall>
</ConsultationCallResponse>
```

10.4.3 Negative Service Response

The CSTAErrorCode message is used to indicate a negative response.

The common elements are:

- A mandatory error category element. There is a choice of seven error categories each with a set of error values. See ECMA-323 clause 9.19 for a list of all possible error codes. Some commonly used error codes are:
  - operation (serviceNotSupported) – UA does not support the service
  - operation (invalidConnectionIdentifier) – the connection identifier in the request is not valid
  - operation (invalidDeviceIdentifier) – the device identifier in the request is not valid
  - availability (resourceBusy) – the UA is not in a state where it can accept a Consultation Call request.

This example shows a negative response because the UA cannot accept a Consultation Call request because it is busy.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <availability>resourceBusy</availability>
</CSTAErrorCode>
```

This example shows a negative response because the Consultation Call service is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>serviceNotSupported</operation>
</CSTAErrorCode>
```

This example shows a negative response because the device identifier in the request is invalid.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>invalidDeviceIdentifier</operation>
</CSTAErrorCode>
```

10.5 Deflect Call

An application uses this service to move an existing call at a UA to another destination.

The UA should provide a positive response if it can successfully deflect the call.

The UA should provide a negative response with an appropriate error code if it cannot successfully deflect the call. Some examples:

- The specified call is not in the appropriate state at the UA.
- The UA cannot uniquely identify the call based upon the connection identifier in the request.
- The UA does not support this service.

10.5.1 Service Request

The common Service Request elements are:

- callToBeDiverted – this mandatory element contains the connection identifier of the call to be diverted. The connection identifier consists of the logical address of the device (e.g. line) and the call identifier of the call. The application uses the contents of the connection element provided by either CSTA events, or by other mechanisms (e.g. SIP event packages).
- newDestination – this mandatory element specifies the device identifier (e.g., SIP URI) where the call will be sent.

In this example, the application indicates that call with the call identifier of 123456789 at the UA sip:tom1@domain.com should be deflected to the address sip:tom2@domain.com.

```
<?xml version="1.0" encoding="UTF-8"?>
<DeflectCall xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <callToBeDiverted>
      <callID>123456789</callID>
      <deviceID>sip:tom1@domain.com</deviceID>
   </ callToBeDiverted >
   <newDestination>sip:tom2@domain.com</ newDestination>
</DeflectCall>
```

10.5.2 Positive Service Response

There are no common Service Response elements in the positive service response.

This example indicates that the SIP phone is deflecting the call at the device.

```
<?xml version="1.0" encoding="UTF-8"?>
<DeflectCallResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"/>
```

10.5.3 Negative Service Response

The CSTAErrorCode message is used to indicate a negative response.

The common elements are:

- A mandatory error category element. There is a choice of seven error categories each with a set of error values. See ECMA-323 clause 9.19 for a list of all possible error codes. Some commonly used error codes are:
  - operation (serviceNotSupported) – the UA does not support the service
  - operation (invalidConnectionIdentifier) – the connection identifier in the request is not valid
  - stateIncompatibility (invalidConnectionState) – call is not in the appropriate state at the UA.

This example shows a negative response because the Deflect Call service is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>serviceNotSupported</operation>
</CSTAErrorCode>
```

This example shows a negative response because the connection identifier in the request is invalid.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>invalidConnectionIdentifier</operation>
</CSTAErrorCode>
```

This example shows a negative response because the connection is not in the correct state.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <stateIncompatibility>invalidConnectionState</stateIncompatibility>
</CSTAErrorCode>
```

10.6 Generate Digits

An application uses this service to cause a series of digits to be sent on an existing call at a UA.

The UA should provide a positive response if it supports the service, the service request is valid, and the call is in the connected state. This means that the UA will attempt to send the digits but does not indicate that this will be successful.

The UA should provide a negative response with an appropriate error code if it will not be attempting to generate the digits:

- The specified call is not in the connected state.
- The UA cannot uniquely identify the call based upon the connection identifier in the request.
- The UA does not support this service.

10.6.1 Service Request

The common Service Request elements are:

- connectionToSendDigits – this mandatory element contains the connection identifier of the call on which the digits will be sent. The connection identifier consists of the logical address of the device (e.g. line) and the call identifier of the call. The application uses the contents of the connection element provided by either CSTA events, or by other mechanisms (e.g. SIP event packages).

- charactersToSend – this mandatory element specifies the string of characters to send. The characters consist of the following (DTMF) characters: 0,1,2,3,4,5,6,7,8,9, *,#, A,B,C,D.

In this example, the application indicates that the digit "1" should be sent on the call with the call identifier of 123456789 at the UA sip:tom1@domain.com.

```
<?xml version="1.0" encoding="UTF-8"?>
<GenerateDigits xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <connectionToSendDigits>
      <callID>123456789</callID>
      <deviceID>sip:tom1@domain.com</deviceID>
   </connectionToSendDigits >
   <charactersToSend>1</charactersToSend>
</GenerateDigits>
```

10.6.2 Positive Service Response

There are no common Service Response elements in the positive service response.

This example indicates that the SIP phone has validated the request and will attempt to generate the digits.

```
<?xml version="1.0" encoding="UTF-8"?>
<GenerateDigitsResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"/>
```

10.6.3 Negative Service Response

The CSTAErrorCode message is used to indicate a negative response.

The common elements are:

- A mandatory error category element. There is a choice of seven error categories each with a set of error values. See ECMA-323 clause 9.19 for a list of all possible error codes. Some commonly used error codes are:

  — operation (serviceNotSupported) – the UA does not support the service

  — operation (invalidConnectionIdentifier) – the connection identifier in the request is not valid

  — operation (invalidParameterValue) – the charactersToSend value in the request is not valid

  — stateIncompatibility (invalidConnectionState) – call is not in the connected state.

This example shows a negative response because the Deflect Call service is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>serviceNotSupported</operation>
```

```
</CSTAErrorCode>
```

This example shows a negative response because the connection identifier in the request is invalid.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>invalidConnectionIdentifier</operation>
</CSTAErrorCode>
```

10.7 Hold Call

An application uses this service to hold a connected call at a UA.

The UA should provide a positive response if it can successfully hold the call.

The UA should provide a negative response with an appropriate error code if it cannot successfully hold the call. Some examples:

- The UA cannot uniquely identify a call based upon the connection identifier in the request.
- The specified call is not in the connected state.
- The UA does not support this service.

10.7.1 Service Request

The common Service Request elements are:

- callToBeHeld – this mandatory element contains the connection identifier of the connection to be held. The connection identifier consists of the logical address of the device (e.g. line) and the call identifier of the call. The application uses the contents of the connection element provided by either CSTA events, or by other mechanisms (e.g. SIP event packages).

In this example, the application indicates that call with the call identifier of 123456789 at the UA sip:tom1@domain.com should be held.

```
<?xml version="1.0" encoding="UTF-8"?>
<HoldCall xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <callToBeHeld>
      <callID>123456789</callID>
      <deviceID>sip:tom1@domain.com</deviceID>
   </callToBeHeld>
</HoldCall>
```

10.7.2 Positive Service Response

There are no common Service Response elements in the positive service response.

This example indicates that the SIP phone is holding the requested call.

```
<?xml version="1.0" encoding="UTF-8"?>
<HoldCallResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"/>
```

10.7.3 Negative Service Response

The CSTAErrorCode message is used to indicate a negative response.

The common elements are:

- A mandatory error category element. There is a choice of seven error categories each with a set of error values. See ECMA-323 clause 9.19 for a list of all possible error codes. Some commonly used error codes are:
  - operation (serviceNotSupported) – the UA does not support the Hold Call service
  - operation (invalidConnectionIdentifier) – the connection identifier in the request is not valid
  - stateIncompatibility (invalidConnectionState) – the call is not in the connected state.

This example shows a negative response because there is no connected call at the UA.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <stateIncompatibility>invalidConnectionState</stateIncompatibility>
</CSTAErrorCode>
```

This example shows a negative response because the Hold Call service is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>serviceNotSupported</operation>
</CSTAErrorCode>
```

This example shows a negative response because the connection identifier in the request is invalid.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>invalidConnectionIdentifier</operation>
</CSTAErrorCode>
```

10.8 Make Call

An application uses the Make Call service to set up a call between the UA and another device.

10.8.1 Service Request

The common Service Request elements are:

- calling Device – this mandatory element specifies the SIP URI of the originating line on the UA.
- calledDevice – this mandatory element specifies the device of the called device.
- autoOriginate – this optional element specifies if the originating UA should either be prompted or not prompted to go manually off hook. The possible values are:
  - prompt (default) – specifies that the originating UA should be prompted (typically the device is rung) to go off hook.

— doNotPrompt – specifies that the originating UA should attempt to connect the originating device without manual interaction (hands free mode). If the UA cannot connect without prompting, the processing of the service should continue and the originating device should be prompted instead.

- userData – this optional element specifies user data that is to be sent to parties in the call. Note that the application should verify that the sending of user data can be supported by the device and by the underlying transport.

In this example, the application wants to establish a call from the originating UA sip:tom1@domain.com (on the SIP phone) to sip:ed@domain.com. The application indicates that the call should be connected to the originating UA without prompting (hands free mode).

```
<?xml version="1.0" encoding="UTF-8"?>
<MakeCall xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <callingDevice>sip:tom1@domain.com</callingDevice>
   <calledDirectoryNumber>sip:ed@domain.com</calledDirectoryNumber>
   <autoOriginate>doNotPrompt</autoOriginate>
</MakeCall>
```

In this example, the application indicates that the originating UA should be prompted to go off hook. Since this is the default value of autoOriginate, it would not need to be included to specify this behaviour. If the originating UA does not go off hook within a UA determined period, the UA should clear the call.

```
<?xml version="1.0" encoding="UTF-8"?>
<MakeCall xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <callingDevice>sip:tom1@domain.com</callingDevice>
   <calledDirectoryNumber>sip:ed@domain.com</calledDirectoryNumber>
   <autoOriginate>prompt</autoOriginate>
</MakeCall>
```

10.8.2 Positive Service Response

This example shows the positive response to the Make Call request.

The common Service Response elements are:

- callingDevice connection identifier – this mandatory element provides the connection identifier that is used to reference the connection in subsequent services. It is also provided in future CSTA events associated with the connection. The connection identifier consists of the logical address of the device (e.g. line) and the call identifier of the call.

In this example, the UA indicates the originating connectionID for the call at the UA.

```
<?xml version="1.0" encoding="UTF-8"?>
<MakeCallResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <callingDevice>
      <callID>123456789</callID>
      <deviceID>sip:tom1@domain.com</deviceID>
   </callingDevice>
</MakeCallResponse>
```

10.8.3 Negative Service Response

The CSTAErrorCode message is used to indicate a negative response.

The common elements are:

- A mandatory error category element. There is a choice of seven error categories each with a set of error values. See ECMA-323 clause 9.19 for a list of all possible error codes. Some commonly used error codes are:

  — operation (serviceNotSupported) – UA does not support the service

  — operation (invalidCallingDeviceIdentifier) – the connection identifier in the request is not valid

  — operation (invalidCalledDeviceIdentifier) – the connection identifier in the request is not valid

  — availability (resourceBusy) – the UA is not in a state where it can accept a Make Call request.

This example shows a negative response because the UA cannot accept a Make Call request because it is busy.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <availability>resourceBusy</availability>
</CSTAErrorCode>
```

This example shows a negative response because the Make Call service is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>serviceNotSupported</operation>
</CSTAErrorCode>
```

This example shows a negative response because the calling device in the request is invalid.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>invalidCallingDeviceIdentifier</operation>
</CSTAErrorCode>
```

10.9 Reconnect Call

An application uses this service to clear a specified connection at the UA and retrieve a specified held connection at the same UA.

The UA should provide a positive response if it can successfully perform the service.

The UA should provide a negative response with an appropriate error code if it cannot successful reconnect the call. Some examples:

- The UA cannot uniquely identify a connected and/or held call based upon the connection identifiers in the request.
- One or both calls at the UA are not in the appropriate state.
- The UA does not support this service.

10.9.1 Service Request

The common Service Request elements are:

- activeCall – this mandatory element contains the connection identifier of the active call to be cleared. The connection identifier consists of the logical address of the device (e.g. line) and the call identifier of the call. The application uses the contents of the connection element provided by either CSTA events, or by other mechanisms (e.g. SIP event packages).

- heldCall – this mandatory element contains the connection identifier of the held call to be retrieved. The connection identifier consists of the logical address of the device (e.g. line) and the call identifier of the call. The application uses the contents of the connection element provided by either CSTA events, or by other mechanisms (e.g. SIP event packages).

In this example, the application wants to clear the active connection and retrieve the held connection at the same UA.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReconnectCall xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
  <activeCall>
    <callID>5678</callID>
    <deviceID>sip:tom1@domain.com</deviceID>
  </activeCall>
  <heldCall>
    <callID>123456789</callID>
    <deviceID>sip:tom1@domain.com</deviceID>
  </heldCall>
</ReconnectCall>
```

10.9.2 Positive Service Response

There are no common Service Response elements in the positive service response.

This example indicates that the UA is clearing the active call and retrieving the held call.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReconnectCallResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"/>
```

10.9.3 Negative Service Response

The CSTAErrorCode message is used to indicate a negative response.

The common elements are:

- A mandatory error category element. There is a choice of seven error categories each with a set of error values. See ECMA-323 clause 9.19 for a list of all possible error codes. Some commonly used error codes are:

  — operation (serviceNotSupported) – UA does not support the Reconnect Call service

  — operation (invalidConnectionIdentifier) – one of the connection identifiers in the request is not valid

  — stateIncompatibility (invalidConnectionState) – one (or both) of the calls is not in the appropriate state.

This example shows a negative response because the Reconnect Call service is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>serviceNotSupported</operation>
</CSTAErrorCode>
```

This example shows a negative response because one of the connection identifiers in the request is invalid.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>invalidConnectionIdentifier</operation>
</CSTAErrorCode>
```

10.10 Retrieve Call

An application uses this service to retrieve a held call at a UA.

The UA should provide a positive response if it can successfully retrieve the call.

The UA should provide a negative response with an appropriate error code if it cannot successful retrieve the call: Some examples:

- The specified call is not in the held state.
- The UA does not support this service.

10.10.1 Service Request

The common Service Request elements are:

- callToBeRetrieved – this mandatory element contains the connection identifier of the connection to be retrieved. The connection identifier consists of the logical address of the device (e.g. line) and the call identifier of the call. The application uses the contents of the connection element provided by either CSTA events, or by other mechanisms (e.g. SIP event packages).

In this example, the application indicates that call with the call identifier of 123456789 at the UA sip:tom1@domain.com should be retrieved.

```
<?xml version="1.0" encoding="UTF-8"?>
<RetrieveCall xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <callToBeRetrieved>
      <callID>123456789</callID>
      <deviceID>sip:tom1@domain.com</deviceID>
   </callToBeRetrieved>
</RetrieveCall>
```

10.10.2 Positive Service Response

There are no common Service Response elements in the positive service response.

This example indicates that the UA is retrieving the requested call.

```
<?xml version="1.0" encoding="UTF-8"?>
<RetrieveCallResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"/>
```

10.10.3 Negative Service Response

The CSTAErrorCode message is used to indicate a negative response.

The common elements are:

- A mandatory error category element. There is a choice of seven error categories each with a set of error values. See ECMA-323 clause 9.19 for a list of all possible error codes. Some commonly used error codes are:

  — operation (serviceNotSupported) – the UA does not support the service

  — operation (invalidConnectionIdentifier) – the connection identifier in the request is not valid

  — stateIncompatibility (invalidConnectionState) – call is not in the held state.

This example shows a negative response because there is no held call at the device.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <stateIncompatibility>invalidConnectionState</stateIncompatibility>
</CSTAErrorCode>
```

This example shows a negative response because the Retrieve Call service is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>serviceNotSupported</operation>
</CSTAErrorCode>
```

This example shows a negative response because the connection identifier in the request is invalid.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>invalidConnectionIdentifier</operation>
</CSTAErrorCode>
```

10.11 Single Step Transfer Call

An application uses this service to transfer an existing connection at a UA to another device.

This transfer is performed in a single step, that is, the UA does not place the existing call on hold before transferring the call.

The UA should provide a positive response if it can successfully transfer the call.

The UA should provide a negative response with an appropriate error code if it cannot successfully transfer the call. Some examples:

- The specified call is not in the appropriate state at the UA.
- The UA cannot uniquely identify the call based upon the connection identifier in the request.
- The UA does not support this service.

10.11.1 Service Request

The common Service Request elements are:

- activeCall – this mandatory element contains the connection identifier of the call to be transferred. The connection identifier consists of the logical address of the device (e.g. line) and the call identifier of the call. The application uses the contents of the connection element provided by either CSTA events, or by other mechanisms (e.g. SIP event packages).

- transferredTo – this mandatory element specifies the device identifier (e.g., SIP URI) where the call will be sent.

In this example, the application indicates that call with the call identifier of 123456789 at the UA sip:tom1@domain.com should be transferred to the phone number +1(800)555-1212.

```
<?xml version="1.0" encoding="UTF-8"?>
<SingleStepTransferCall xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
  <activeCall>
        <callID>123456789</callID>
        <deviceID>sip:tom1@domain.com</deviceID>
  </activeCall>
  <transferredTo>+1(800)555-1212</transferredTo>
</SingleStepTransferCall>
```

10.11.2 Positive Service Response

There are no common Service Response elements in the positive service response.

This example indicates that the SIP phone is transferring the call at the device.

```
<?xml version="1.0" encoding="UTF-8"?>
<DeflectCallResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"/>
```

10.11.3 Negative Service Response

The CSTAErrorCode message is used to indicate a negative response.

The common elements are:

- A mandatory error category element. There is a choice of seven error categories each with a set of error values. See ECMA-323 clause 9.19 for a list of all possible error codes. Some commonly used error codes are:

  — operation (serviceNotSupported) – the UA does not support the service

  — operation (invalidConnectionIdentifier) – the connection identifier in the request is not valid

  — stateIncompatibility (invalidConnectionState) – call is not in the appropriate state at the UA.

This example shows a negative response because the Single Step Transfer Call service is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>serviceNotSupported</operation>
</CSTAErrorCode>
```

This example shows a negative response because the connection identifier in the request is invalid.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>invalidConnectionIdentifier</operation>
</CSTAErrorCode>
```

This example shows a negative response because the connection is not in the correct state.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <stateIncompatibility>invalidConnectionState</stateIncompatibility>
</CSTAErrorCode>
```

10.12 Transfer Call

An application uses this service to transfer a call held at the UA to an active call at the same UA.

The held and active calls at the UA are merged into a new call and the UA is released from the call.

The UA should provide a positive response if it can successfully transfer the call.

The UA should provide a negative response with an appropriate error code if it cannot successfully transfer the call. Some examples:

- The specified call is not in the appropriate state at the UA.
- The UA cannot uniquely identify the call based upon the connection identifier in the request.
- The UA does not support this service.

10.12.1 Service Request

The common Service Request elements are:

- heldCall – this mandatory element contains the connection identifier of the held call at the UA. The connection identifier consists of the logical address of the device (e.g. line) and the call identifier of the call. The application uses the contents of the connection element provided by either CSTA events, or by other mechanisms (e.g. SIP event packages).
- activeCall – connection identifier of the active call at the UA. The connection identifier consists of the logical address of the device (e.g. line) and the call identifier of the call. The application uses the contents of the connection element provided by either CSTA events, or by other mechanisms (e.g. SIP event packages).

In this example, the application indicates that calls with the call identifier of 123456789 and 5678 at the UA sip:tom1@domain.com should be merged into a new call and the UA sip:tom1@domain.com is released from the resulting call.

```
<?xml version="1.0" encoding="UTF-8"?>
<TransferCall xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
 <heldCall>
  <callID>5678</callID>
  <deviceID>sip:tom1@domain.com</deviceID>
 </activeCall>
 <activeCall>
```

```
    <callID>123456789</callID>
    <deviceID>sip:tom1@domain.com</deviceID>
  </heldCall>
</TransferCall>
```

10.12.2 Positive Service Response

This example shows the positive response to the Transfer Call request.

The common Service Response elements are:

- transferredCall connection identifier – this mandatory element provides the connection identifier of the transferredTo device in the resulting call. The connection identifier consists of the logical address of the device (e.g. line) and the call identifier of the call.

In this example, the UA indicates the connectionID of the transferred call at the UA sip:jane1@domain.com.

```
<?xml version="1.0" encoding="UTF-8"?>
<TransferCallResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
  <transferredCall>
    <callID>123456789</callID>
    <deviceID>sip:jane1@domain.com</deviceID>
  </transferredCall>
</TransferCallResponse>
```

10.12.3 Negative Service Response

The CSTAErrorCode message is used to indicate a negative response.

The common elements are:

- A mandatory error category element. There is a choice of seven error categories each with a set of error values. See ECMA-323 clause 9.19 for a list of all possible error codes. Some commonly used error codes are:

  — operation (serviceNotSupported) – the UA does not support the service

  — operation (invalidConnectionIdentifier) – the connection identifier in the request is not valid

  — stateIncompatibility (invalidConnectionState) – call is not in the appropriate state at the UA.

This example shows a negative response because the Transfer Call service is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
  <operation>serviceNotSupported</operation>
</CSTAErrorCode>
```

This example shows a negative response because the connection identifier in the request is invalid.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
  <operation>invalidConnectionIdentifier</operation>
</CSTAErrorCode>
```

This example shows a negative response because the connection is not in the correct state.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <stateIncompatibility>invalidConnectionState</stateIncompatibility>
</CSTAErrorCode>
```

11 Physical Phone Features

This clause shows examples of how CSTA can be used to control and query physical components commonly found on a UA like a speakerphone, microphone, and a message waiting indicator.

In addition to these components, ECMA-269 and ECMA-323 specifies additional features to control physical attributes such as buttons, displays, hook switch status, lamps, and ringer status.

Speakers and Microphone Components

An *auditory apparatus* is a component that contains a speaker and/or a microphone. A UA can support several auditory apparatuses associated with it such as:

- Handset – an auditory apparatus that is held in a person's hand that contains a microphone and speaker.
- Headset - an auditory apparatus that is worn on a person's head and contains a microphone and speaker.
- Speakerphone - an auditory apparatus that does not require a person's body to use the apparatus and contains a microphone and speaker.
- Speaker-only speakerphone – a speakerphone without a microphone.
- Microphone-only speakerphone – a speakerphone without a speaker.

Each auditory apparatus has an identifier, called an auditoryApparatusID, that allows an application to control attributes associated with a specific auditory apparatus such as:

- Microphone gain – the input level for the microphone.
- Microphone mute – temporarily disables the microphone.
- Speaker Volume – the output level for the speaker.
- Speaker Mute – temporarily disables the speaker output.

Addressing a Physical Element of a UA

The physical element of a UA is referenced through a deviceID (SIP URI) which, for single line phones, is typically the same SIP URI used to reference a UA's logical element (line).

For uaCSTA, since the CSTA request is sent on an existing SIP dialog, the physical element referenced will be that which terminates the dialog.

11.1 Get Message Waiting Indicator

An application can use this service to query a UA's message waiting indicator status.

The message waiting indicator can be used to notify a user (typically via a dedicated lamp on a phone) when messages are available.

11.1.1 Service Request

The common Service Request elements are:

- device – this mandatory element specifies the URI of the UA.

In this example, an application requests the status of a UA's message waiting indicator.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetMessageWaitingIndicator xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <device>sip:tom1@domain.com</device>
</GetMessageWaitingIndicator>
```

11.1.2 Service Response

The common Service Response elements are:

- messageWaitingOn – this mandatory element specifies the value of the message waiting indicator. The possible values are:
  - true – message waiting in ON
  - false – message waiting is OFF.

In this example, the UA indicates that the message waiting indicator is on.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetMessageWaitingIndicatorResponse
   xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <messageWaitingOn>true</messageWaitingOn>
   </GetMessageWaitingIndicatorResponse>
```

11.2 Set Message Waiting Indicator

An application can use this service to control a UA's message waiting indicator status.

The message waiting indicator can be used to notify a user (typically via a dedicated lamp on a phone) when messages are available.

11.2.1 Service Request

The common Service Request elements are:

- device – specifies the URI of the UA.
- messageWaitingOn – this mandatory element specifies the requested setting of the message waiting feature. The possible values are:
  - true – message waiting in ON
  - false – message waiting is OFF.

Example of a request from an application to clear the message waiting indicator:

```
<?xml version="1.0" encoding="UTF-8"?>
<SetMessageWaitingIndicator xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <device>sip:tom1@domain.com</device>
   <messageWaitingOn>false</messageWaitingOn>
</SetMessageWaitingIndicator>
```

11.2.2 Service Response

There are no common Service Response elements.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
  <SetMessageWaitingIndicatorResponse
xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"/>
```

11.3 Get Speaker Mute

An application can use this service to obtain a UA's speaker mute setting.

11.3.1 Service Request

The common Service Request elements are:

- device – this mandatory element specifies the URI of the phone.
- auditoryApparatus – this optional element specifies the phone's auditory apparatus associated with the speaker. If this element is not provided, the phone will provide a list of the mute status associated with all of the auditoryApparatuses.

Example of a request from an application to get the phone's speaker mute setting for auditoryApparatus 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetSpeakerMute xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <device>sip:tom1@domain.com</device>
   <auditoryApparatus>1</auditoryApparatus>
</GetSpeakerMute>
```

11.3.2 Service Response

The common Service Response elements are:

- speakerMuteList – this mandatory element specifies a list of speakerMuteItem elements that contain:
  - auditoryApparatus – specifies the phone's auditory apparatus
  - speakerMuteOn – specifies the speaker mute setting. The possible values are: true (mute is ON) and false (mute is off).

This example response shows that the speaker associated with auditory apparatus 1 is set Mute on.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetSpeakerMuteResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <speakerMuteList>
      <speakerMuteItem>
         <auditoryApparatus>1</auditoryApparatus>
         <speakerMuteOn>true</speakerMuteOn>
      </speakerMuteItem>
   </speakerMuteList>
</GetSpeakerMuteResponse>
```

11.4 Set Speaker Mute

An application can use this service to control a UA's speaker mute status.

11.4.1 Service Request

The common Service Request elements are:

- device – this mandatory element specifies the URI of the UA.
- auditoryApparatus – this mandatory element specifies the UA's auditory apparatus that contains the speaker to be controlled
- speakerMuteOn – this mandatory element specifies the requested setting of the speaker mute. The possible values are:
  - true – Mute is ON
  - false – Mute is OFF.

Example of a request from an application to set the UA's speaker mute status to Mute:

```
<?xml version="1.0" encoding="UTF-8"?>
<SetSpeakerMute xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <device>sip:tom@domain.com</device>
   <auditoryApparatus>1</auditoryApparatus>
   <speakerMuteOn>true</speakerMuteOn>
</SetSpeakerMute>
```

11.4.2 Service Response

There are no common Service Response elements.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
  <SetSpeakerMuteResponse
xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"/>
```

11.5 Get Speaker Volume

An application can use this service to obtain a UA's speaker volume setting.

11.5.1 Service Request

The common Service Request elements are:

- device – this mandatory element specifies the URI of the UA.
- auditoryApparatus – this optional element specifies the UA's auditory apparatus associated with the speaker. If this element is not provided, the UA will provide a list of the volume setting associated with all of the auditoryApparatuses.

Example of a request from an application to get the UA's speaker volume setting. In this case, the application is omitting a auditoryApparatus element so the UA should provide a list of all of its auditoryApparatuses.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetSpeakerVolume xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <device>sip:tom1@domain.com</device>
</GetSpeakerVolume>
```

11.5.2 Service Response

The common Service Response elements are:

- speakerVolumeList – this mandatory element specifies a list of the UA's auditory apparatuses:
  - auditoryApparatus – this mandatory element specifies the UA's auditory apparatus that contains the volume setting
  - speakerVolAbs – this optional element specifies the absolute volume setting of the speaker where the setting of 0 indicates the most silent and the value of 100 indicates the maximum possible value. This element is not provided when the UA cannot provide the speaker volume.

This example response shows that the speaker associated with auditory apparatus 1 is set to maximum volume. Although there is another auditory apparatus on the UA, its speaker volume cannot be provided.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetSpeakerVolumeResponse xmlns="http://www.ecma-international.org/standards/ecma-
323/csta/ed3">
   <speakerVolumeList>
      <speakerVolumeItem>
         <auditoryApparatus>1</auditoryApparatus>
         <speakerVolAbs>100</speakerVolAbs>
      </speakerVolumeItem>
      <speakerVolumeItem>
         <auditoryApparatus>2</auditoryApparatus>
      </speakerVolumeItem>
   </speakerVolumeList>
</GetSpeakerVolumeResponse>
```

11.6 Set Speaker Volume

An application can use this service to control a UA's speaker volume setting.

11.6.1 Service Request

The common Service Request elements are:

- device – this mandatory element specifies the URI of the UA.
- auditoryApparatus – this mandatory element specifies the UA's auditory apparatus that contains the speaker to be controlled.
- speakerVolume – this mandatory element specifies either a speaker volume as an absolute value or that the volume should be incremented or decremented by a phone-specified amount (typically corresponding to an amount that would be adjusted by manually pressing the volume button on a phone). One of the following is provided:
  - volAbs – is used to provide an absolute volume setting value where the setting of 0 indicates the most silent and the value of 100 indicates the maximum possible value.
  - volInc – specifies the value "Increment" if the volume should be incremented or "Decrement" if the value should be decremented.

Example of a request from an application to set the phone's speaker volume to maximum volume:

```
<?xml version="1.0" encoding="UTF-8"?>
<SetSpeakerVolume xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <device>sip:tom1@domain.com</device>
   <auditoryApparatus>1</auditoryApparatus>
   <speakerVolume>
      <volAbs>100</volAbs>
   </speakerVolume>
</SetSpeakerVolume>
```

Example of a request from an application to decrement the phone's speaker volume:

```
<?xml version="1.0" encoding="UTF-8"?>
<SetSpeakerVolume xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <device>sip:tom1@domain.com</device>
   <auditoryApparatus>1</auditoryApparatus>
   <speakerVolume>
      <volInc>decrement</volInc>
   </speakerVolume>
</SetSpeakerVolume>
```

11.6.2 Service Response

There are no common Service Response elements.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
  <SetSpeakerVolumeResponse
xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"/>
```

12 Logical Phone Features

A logical element of a UA refers to the part of the UA that manages and interacts with calls.

A logical element of a UA is referenced through a deviceID (SIP URI). A UA may support one or more addressable logical elements (e.g. multi line phones). A UA may share a logical element with other physical devices (e.g. bridged appearance phone).

This clause shows examples of how CSTA can be used to control and query attributes associated with a logical element such as do not disturb and forwarding. In addition to these features, ECMA-269 and ECMA-323 specify many additional features to control logical attributes of a UA.

12.1 Get Do Not Disturb

An application can use this service to query a UA's do not disturb status.

The do not disturb feature is used to prevent incoming calls at a UA.

12.1.1 Service Request

The common Service Request elements are:

- device – this mandatory element specifies the URI of the UA.

In this example, an application requests the status of a UA's do not disturb status.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetDoNotDisturb xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
        <device>sip:tom1@domain.com</device>
</GetDoNotDisturb>
```

12.1.2 Service Response

The common Service Response elements are:

- doNotDisturb – this mandatory element specifies the value of the message waiting indicator. The possible values are:
  - true – do not disturb feature is ON
  - false – do not disturb feature is OFF.

In this example, the UA indicates that the do not disturb status is off.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetDoNotDisturbResponse xmlns="http://www.ecma-international.org/standards/ecma-
323/csta/ed3">
    <doNotDisturbOn>false</doNotDisturbOn>
</GetDoNotDisturbResponse>
```

12.2 Set Do Not Disturb

An application can use this service to control a UA's do not disturb status.

The do not disturb status can be used to prevent incoming calls at a UA.

12.2.1 Service Request

The common Service Request elements are:

- device – this mandatory element specifies the URI of the phone.
- doNotDisturbOn – this mandatory element specifies the requested setting of the do not disturb feature. The possible values are:
  - true – do not disturb is ON
  - false – do not disturb is OFF.

Example of a request from an application to set the do not disturb status to ON:

```
<?xml version="1.0" encoding="UTF-8"?>
<SetDoNotDisturb xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
        <device>sip:tom1@domain.com</device>
        <doNotDisturbOn>true</doNotDisturbOn>
</SetDoNotDisturb>
```

12.2.2 Service Response

There are no common Service Response elements.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SetDoNotDisturbResponse xmlns="http://www.ecma-international.org/standards/ecma-
323/csta/ed3"/>
```

12.3 Get Forwarding

An application can use this service to query the current status of the forwarding feature of a UA.

The forwarding feature is used to redirect incoming calls to an alternate destination.

12.3.1 Service Request

The common Service Request elements are:

- device – this mandatory element specifies the URI of the UA.

In this example, an application requests the status of a UA's forwarding feature.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetForwarding xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
        <device>sip:tom1@domain.com</device>
</GetForwarding>
```

12.3.2 Service Response

The common Service Response elements are:

- forwardingList – this mandatory element specifies a list of forwardListItem elements that contain:

— forwardingType – this mandatory element indicates the forwarding types. Some of the common forwarding types are: forwardImmediate, forwardBusy, forwardDND, and forwardNoAns.

— forwardStatus – this mandatory element indicates the forwarding status of false (the forwarding type is not active) or true (the forwarding type is active).

— forwardDN – this mandatory element specifies the destination to which calls are forwarded.

— ringCount – this mandatory element specifies the number of ring cycles prior to forwardNoAns.

In this example, the phone indicates that the UA is set to forward all incoming calls to sip:tom2@domain.com.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetForwardingResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
  <forwardingList>
    <forwardListItem>
      <forwardingType>forwardImmediate</forwardingType>
      <forwardStatus>true</forwardStatus>
      <forwardDN>sip:tom2@domain.com</forwardDN>
    </forwardListItem>
  </forwardingList>
</GetForwardingResponse>
```

In this example, the UA indicates that it will forward incoming calls to sip:tom2@domain.com if the phone is not answered in 10 ring cycles.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetForwardingResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3 ">
  <forwardingList>
    <forwardListItem>
      <forwardingType>forwardNoAns</forwardingType>
      <forwardStatus>true</forwardStatus>
      <forwardDN>sip:tom2@domain.com</forwardDN>
      <ringCount>10</ringCount>
    </forwardListItem>
  </forwardingList>
</GetForwardingResponse>
```

12.4 Set Forwarding

An application can use this service to control a UA's forwarding feature.

The forwarding feature is used to redirect incoming calls to an alternate destination.

12.4.1 Service Request

The common Service Request elements are:

- device – this mandatory element specifies the URI of the UA.
- forwardingType – this mandatory element specifies the requested forwarding type. Some of the common forwarding types are: forwardImmediate, forwardBusy, forwardDND, and forwardNoAns.
- activateForward – this mandatory element specifies the forwarding status of false (the forwarding type is not active) or true (the forwarding type is active).

- forwardDN – this mandatory element specifies the destination to which calls are forwarded.
- ringCount – this optional element specifies the number of ring cycles prior to forwardNoAns.

Example of a request from an application to set the UA to forward unanswered calls to UA sip:tom2@domain.com after 10 ring cycles:

```
<?xml version="1.0" encoding="UTF-8"?>
<SetForwarding xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <device>sip:tom1@domain.com</device>
      <forwardingType>forwardNoAns</forwardingType>
      <activateForward>true</activateForward>
      <forwardDN>sip:tom2@domain.com</forwardDN>
      <ringCount>10</ringCount>
</SetForwarding>
```

12.4.2 Service Response

There are no common Service Response elements.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<SetForwardingResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"/>
```

13 Monitoring Services and Events

13.1 Monitor Start

An application uses the Monitor Start service to observe changes in the state of connections and features at a UA.

The UA should provide a positive response if it can establish a monitor. The UA provides a monitorCrossRefID in the positive response that is used to correlate subsequent CSTA events to this Monitor Start request.

The UA should provide a negative response with an appropriate error code if it cannot successfully establish a monitor. Some examples:

- The UA does not recognize the monitorObject in the service request.
- The UA is not permitted to establish a monitor for the monitorObject in the service request.
- The UA does not support this service.

13.1.1 Service Request

The common Service Request elements are:

- monitorObject – this mandatory element specifies the URI of the UA to be observed.
- monitorType – this optional element specifies the type of monitor. If not provided in the service request the UA uses a monitorType of “device” as the default value.

- requestedMonitorMediaClass – this optional element specifies a list of the type of media that the application in interested in observing. If not provided in the service request the UA should assume that the application is interested in receiving events for all CSTA media types supported by the UA.

In this example, an application requests that the UA sip:tom@sip.com be monitored. The monitorType is device which indicates that only calls at the specified monitorObject are monitored. The application is interested in observing both voice and IM calls at the specified device as shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<MonitorStart xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <monitorObject>
      <deviceObject>sip:tom@sip.com</deviceObject>
   </monitorObject>
   <monitorType>device</monitorType>
   <requestedMonitorMediaClass>
      <voice>true</voice>
      <im>true</im>
   </requestedMonitorMediaClass>
</MonitorStart>
```

13.1.2 Positive Service Response

This example shows the positive response to the Monitor Start request.

The common Service Response elements are:

- monitorCrossRefID – this element is a unique value that that is used to correlate subsequent CSTA events to the monitor request that initiated the event reporting.

- actualMonitorMediaClass – this optional element specifies a list of the actual types of media that the UA is observing. The types of media observed by the UA may be the same or a subset of the media types requested in the service request. The UA may omit this element if the actualMonitorMediaClass type is the same as what was requested in the service request.

In this example, the UA indicates it has assigned a value of 5665621 to the monitorCrossRefID associated with this monitor. Subsequent CSTA events generated as a result of this monitor request will contain this value. The UA is also indicating that both voice and Instant Messaging media class types are observed for this monitor.

```
<?xml version="1.0" encoding="UTF-8"?>
<MonitorStartResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <monitorCrossRefID>5665621</monitorCrossRefID>
   <actualMonitorMediaClass>
      <voice>true</voice>
      <im>true</im>
   </actualMonitorMediaClass>
</MonitorStartResponse>
```

13.1.3 Negative Service Response

The CSTAErrorCode message is used to indicate a negative response.

The common elements are:

- A mandatory error category element. There is a choice of seven error categories each with a set of error values. See ECMA-323 clause 9.19 for a list of all possible error codes. Some commonly used error codes are:

  — operation (serviceNotSupported) – UA does not support the service

  — operation (invalidMonitorObject) – the monitor object in the request is not valid

  — operation (privilegeViolationSpecifiedDevice) – the UA could not establish the monitor because it does not have the authorization to do so

  — system resource availability (outOfService) – the UA cannot start a monitor because the monitor object is out of service.

This example shows a negative response because the UA is not authorized to observe this monitor object.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
  <operation>privilegeViolationSpecifiedDevice</operation>
</CSTAErrorCode>
```

This example shows a negative response because the Monitor Start service is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>serviceNotSupported</operation>
</CSTAErrorCode>
```

This example shows a negative response because the monitor object in the request is invalid.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>invalidMonitorObject</operation>
</CSTAErrorCode>
```

13.2 Monitor Stop

An application uses the Monitor Stop service to stop an existing monitor.

The UA should provide a negative response with an appropriate error code if it cannot stop the specified monitor. Some examples:

- The UA does not recognize the monitorCrossRefID in the service request.
- The UA does not support this service.

13.2.1 Service Request

The common Service Request elements are:

- monitorCrossRefID – this mandatory element specifies the monitor cross reference ID that was provided in the Monitor Stop positive response.

In this example, an application requests that the existing monitor associated with the monitor cross reference identifier of 5665621 be stopped.

```
<?xml version="1.0" encoding="UTF-8"?>
<MonitorStop xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <monitorCrossRefID>5665621</monitorCrossRefID>
</MonitorStop>
```

13.2.2 Positive Service Response

There are no common Service Response elements in the positive service response.

This example indicates that the UA has stopped the monitor specified in the service request.

```
<?xml version="1.0" encoding="UTF-8"?>
<MonitorStopResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
```

13.2.3 Negative Service Response

The CSTAErrorCode message indicates a negative response.

The common elements are:

- A mandatory error category element. There is a choice of seven error categories each with a set of error values. See ECMA-323 clause 9.19 for a list of all possible error codes. Some commonly used error codes are:

  — operation (serviceNotSupported) – UA does not support the service

  — operation (invalidMonitorCrossRefID) – the monitor object in the request is not valid.

This example shows a negative response because the Monitor Stop service is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>serviceNotSupported</operation>
</CSTAErrorCode>
```

This example shows a negative response because the monitor cross reference identifier in the request is invalid.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>invalidMonitorCrossRefID</operation>
</CSTAErrorCode>
```


13.3 Events

CSTA events indicate changes in the state of a CSTA connection or a feature at a UA.

Events are organized in CSTA standards into categories just like CSTA services.

Clause 8 summarizes the events that are part of the uaCSTA profiles. There are numerous examples of how events are used in Clause 16.

Refer to ECMA-269 for a complete description of all of the CSTA events, event elements, and behaviours.

14 Snapshot Services

14.1 Snapshot Device

An application uses the Snapshot Device service to obtain information about the CSTA connections at a UA.

The UA should provide a positive response with the list of zero or more connections and information about each connection at the UA.

The UA should provide a negative response with an appropriate error code if it cannot provide a positive response. Some examples:

- The UA does not recognize the snapshotObject in the service request.
- The UA is not permitted to snapshot the snapshotObject in the service request.
- The UA does not support this service.

14.1.1 Service Request

The common Service Request elements are:

- snapshotObject – this mandatory element specifies the URI of the UA to be snapshot.

In this example, an application requests that the UA sip:tom@sip.com be snapshot.

```
<?xml version="1.0" encoding="UTF-8"?>
<SnapshotDevice xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <snapshotObject>sip:tom1@domain.com</snapshotObject>
</SnapshotDevice>
```

14.1.2 Positive Service Response

Since the response information is returned in a single response message the element snapshotData is provided with zero or more snapshotDeviceResponseInfo elements (i.e.; corresponding to the number of CSTA connections at the UA).

The common elements in the snapshotDeviceResponseInfo element are:

- connectionIdentifier – this mandatory element provides the connectionID of the connection. This is the connectionID that is used in CSTA services that are applied to the connection.
- localCallState – this mandatory element specifies a compoundCallState which consists of one or more CSTA connection states. The first connection state in the list is the “local” connection state of the connection being reported. Other connection states that reflect other connections in the same call (at different devices) may also be provided, if known to the UA.
- servicesPermitted – this optional parameter indicates what CSTA services can be applied to the connection being reported. Since the call model and features supported by a UA can vary, the UA is encouraged to provide this information to applications so that applications can know which services can be applied to the connection given the state of the connection and the state of the other connections at the UA.
- mediaCallCharacteristics – this optional element indicates the mediaClass (voice, image, IM, etc.) of the connection being reported.

In this example, the UA indicates that there is one CSTA connection at the device with a callID of 32387 and a deviceID of sip:tom1@domain.com. The connection state is alerting. The UA is indicating that three CSTA services can be applied to the connection: Deflect Call, Answer Call, and Clear Connection. The UA indicates that this is a voice call.

```
<?xml version="1.0" encoding="UTF-8"?>
<SnapshotDeviceResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <crossRefIDorSnapshotData>
      <snapshotData>
         <snapshotDeviceResponseInfo>
            <connectionIdentifier>
               <callID>32387</callID>
               <deviceID>sip:tom1@domain.com</deviceID>
            </connectionIdentifier>
            <localCallState>
               <compoundCallState>
                  <localConnectionState>alerting</localConnectionState>
               </compoundCallState>
            </localCallState>
            <servicesPermitted>
               <callControlServices>
                  <answerCall>true</answerCall>
                  <clearConnection>true</clearConnection>
                  <deflectCall>true</deflectCall>
               </callControlServices>
               <callAssociatedServices/>
               <mediaAttachementServices/>
               <routeingServices/>
               <voiceUnitServices/>
            </servicesPermitted>
            <mediaCallCharacteristics>
               <mediaClass>
                  <voice>true</voice>
               </mediaClass>
            </mediaCallCharacteristics>
         </snapshotDeviceResponseInfo>
      </snapshotData>
   </crossRefIDorSnapshotData>
</SnapshotDeviceResponse>
```

In the next example, the UA indicates that there are two (voice media) CSTA connections at the device. The connection state of the first call is held and is connected for the second connection.

The UA is indicating that only one CSTA service can be applied to the first connection: Clear Connection. The UA indicates that the following CSTA services can be applied to the second connection: Transfer Call, Alternate Call, and Reconnect Call.

```
<?xml version="1.0" encoding="UTF-8"?>
<SnapshotDeviceResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <crossRefIDorSnapshotData>
      <snapshotData>
         <snapshotDeviceResponseInfo>
            <connectionIdentifier>
               <callID>32387</callID>
               <deviceID>sip:tom1@domain.com</deviceID>
            </connectionIdentifier>
            <localCallState>
```

```
                <compoundCallState>
                    <localConnectionState>hold</localConnectionState>
                </compoundCallState>
            </localCallState>
            <servicesPermitted>
                <callControlServices>
                    <clearConnection>true</clearConnection>
                </callControlServices>
                <callAssociatedServices/>
                <mediaAttachementServices/>
                <routeingServices/>
                <voiceUnitServices/>
            </servicesPermitted>
            <mediaCallCharacteristics>
                <mediaClass>
                    <voice>true</voice>
                </mediaClass>
            </mediaCallCharacteristics>
        </snapshotDeviceResponseInfo>
        <snapshotDeviceResponseInfo>
            <connectionIdentifier>
                <callID>32388</callID>
                <deviceID>sip:tom1@domain.com</deviceID>
            </connectionIdentifier>
            <localCallState>
                <compoundCallState>
                    <localConnectionState>connected</localConnectionState>
                </compoundCallState>
            </localCallState>
            <servicesPermitted>
                <callControlServices>
                    <alternateCall>true</alternateCall>
                    <reconnectCall>true</reconnectCall>
                    <transferCall>true</transferCall>
                </callControlServices>
                <callAssociatedServices/>
                <mediaAttachementServices/>
                <routeingServices/>
                <voiceUnitServices/>
            </servicesPermitted>
            <mediaCallCharacteristics>
                <mediaClass>
                    <voice>true</voice>
                </mediaClass>
            </mediaCallCharacteristics>
        </snapshotDeviceResponseInfo>
    </snapshotData>
  </crossRefIDorSnapshotData>
</SnapshotDeviceResponse>
```

In the next example, the UA indicates that there are no CSTA connections at the snapshot object.

```
<?xml version="1.0" encoding="UTF-8"?>
<SnapshotDeviceResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <crossRefIDorSnapshotData>
    <snapshotData/>
  </crossRefIDorSnapshotData>
</SnapshotDeviceResponse>
```

14.1.3 Negative Service Response

The CSTAErrorCode message indicates a negative response.

The common elements are:

- A mandatory error category element. There is a choice of seven error categories each with a set of error values. See ECMA-323 clause 9.19 for a list of all possible error codes. Some commonly used error codes are:

  — operation (serviceNotSupported) – UA does not support the service

  — operation (invalidSnapshotObject) – the snapshot object in the request is not valid

  — operation (privilegeViolationSpecifiedDevice) – the UA could not snapshot the object because it does not have the authorization to do so.

This example shows a negative response because the UA is not authorized to snapshot this object.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
    <operation>privilegeViolationSpecifiedDevice</operation>
</CSTAErrorCode>
```

This example shows a negative response because the Snapshot Device service is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
  <operation>serviceNotSupported</operation>
</CSTAErrorCode>
```

This example shows a negative response because the snapshot object in the request is invalid.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
  <operation>invalidMonitorObject</operation>
</CSTAErrorCode>
```


15 Discovery and System Status Services

15.1 Get CSTA Features

This service obtains the list of supported CSTA features (CSTA Services and Events).

This is a system level (i.e. CSTA switching function level) request. If there is only one UA in the switching function, then this service returns all of the features supported by that UA. If there are more than one UA supported in the switching function then this returns the superset of all services and events supported by all of the UAs.

Note that this is a “lightweight” CSTA Capability Exchange service. Other CSTA services can be used to obtain additional information about a specific UA (device).

15.1.1 Service Request

There are no common Service Request elements in the service request.

In this example, an application requests the supported CSTA features.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCSTAFeatures xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
```

15.1.2 Service Response

The common Service Response elements are:

- supportedServices – this mandatory element specifies a list of the supported CSTA services. The services are organized by categories. If the service is not included in the list it is not supported.
- supportedEvents – this mandatory element specifies a list of the supported CSTA events. The events are organized by categories. If the event is not included in the list it is not supported.

In this example, the following services and events are supported by one or more UAs in the switching function.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCSTAFeaturesResponse xmlns="http://www.ecma-international.org/standards/ecma-
323/csta/ed3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <supportedServices>
    <capExchangeServList>
      <getSwitchingFunctionCaps/>
      <getSwitchingFunctionDevices/>
    </capExchangeServList>
    <systemStatServList>
      <requestSystemStatus/>
      <systemStatus/>
    </systemStatServList>
    <monitoringServList>
      <monitorStart/>
      <monitorStop/>
    </monitoringServList>
    <snapshotServList>
      <snapshotDevice/>
    </snapshotServList>
    <callControlServList>
      <answerCall/>
      <clearConnection/>
      <deflectCall/>
      <holdCall/>
      <makeCall/>
      <retrieveCall/>
      <singleStepTransfer/>
    </callControlServList>
  </supportedServices>
  <supportedEvents>
    <callControlEvtsList>
      <connectionCleared/>
      <delivered/>
      <diverted/>
      <established/>
      <failed/>
```

```
            <held/>
            <netwReached/>
            <retrieved/>
            <serviceInitiated/>
            <transferred/>
        </callControlEvtsList>
    </supportedEvents>
</GetCSTAFeaturesResponse>
```

15.1.3 Negative Service Response

The CSTAErrorCode message indicates a negative response.

The common elements are:

- A mandatory error category element. There is a choice of seven error categories each with a set of error values. See ECMA-323 clause 9.19 for a list of all possible error codes. Some commonly used error codes are:

  — operation (serviceNotSupported) – UA does not support the service.

This example shows a negative response because the Get CSTA Features service is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>serviceNotSupported</operation>
</CSTAErrorCode>
```

15.2 Request System Status

This service is used by the application to obtain the status of the system (i.e. application association).

15.2.1 Service Request

There are no common Service Request elements in the service request.

In this example, an application requests the supported CSTA features.

```
<?xml version="1.0" encoding="UTF-8"?>
<RequestSystemStatus xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
```

15.2.2 Service Response

The common Service Response elements are:

- systemStatus – this is the system status. Some of the typical values are:

  — normal – the system status is normal

  — disabled – the system status is disabled. Existing monitors have been lost and will need to be re-established once the system status is no longer disabled.

The following example illustrates how a system status of normal is returned.

```
<?xml version="1.0" encoding="UTF-8"?>
<RequestSystemStatusResponse xmlns="http://www.ecma-international.org/standards/ecma-
323/csta/ed3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <systemStatus>normal</systemStatus>
</RequestSystemStatusResponse>
```

15.2.3 Negative Service Response

The CSTAErrorCode message indicates a negative response.

The common elements are:

- A mandatory error category element. There is a choice of seven error categories each with a set of error values. See ECMA-323 clause 9.19 for a list of all possible error codes. Some commonly used error codes are:

  — operation (serviceNotSupported) – UA does not support the service.

This example shows a negative response because the Request System Status service is not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
<CSTAErrorCode xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <operation>serviceNotSupported</operation>
</CSTAErrorCode>
```

15.3 System Status

This service is used to send the status of the system (i.e. application association) to the application.

15.3.1 Service Request

The common Service Request elements are:

- systemStatus – this mandatory element specifies the status of the system. Some of the common system status values reported are:

  — disabled – all monitors that have been started on the application session have been lost and if further monitoring is needed, the application must re-establish the monitors using the Monitor Start service once the system status is no longer disabled.

  — enabled – all monitors that have been started on the application session have been lost and further monitoring is needed, the application must re-establish the monitors using the Monitor Start service.

  — normal – the system status is normal.

In this example, the switching function is reporting a systemStatus of enabled. This means that any monitors started during the application session have been lost and need to be re-established via the Monitor Start service.

```
<?xml version="1.0" encoding="UTF-8"?>
<SystemStatus xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <systemStatus>enabled</systemStatus>
</SystemStatus>
```

15.3.2 Positive Service Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SystemStatusResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
```

15.3.3 Negative Service Response

There should be no expected negative service response to the System Status request.

16 ECMA-323 Illustrative Examples

This clause provides examples of ECMA-323 XML messages when used with SIP. The clause is organized into sections that address the environments discussed in clause 5.

16.1 Controlling a SIP UA

This section shows examples of how uaCSTA (CSTA over SIP) can be used to control a SIP UA that, in this example, is a SIP phone.

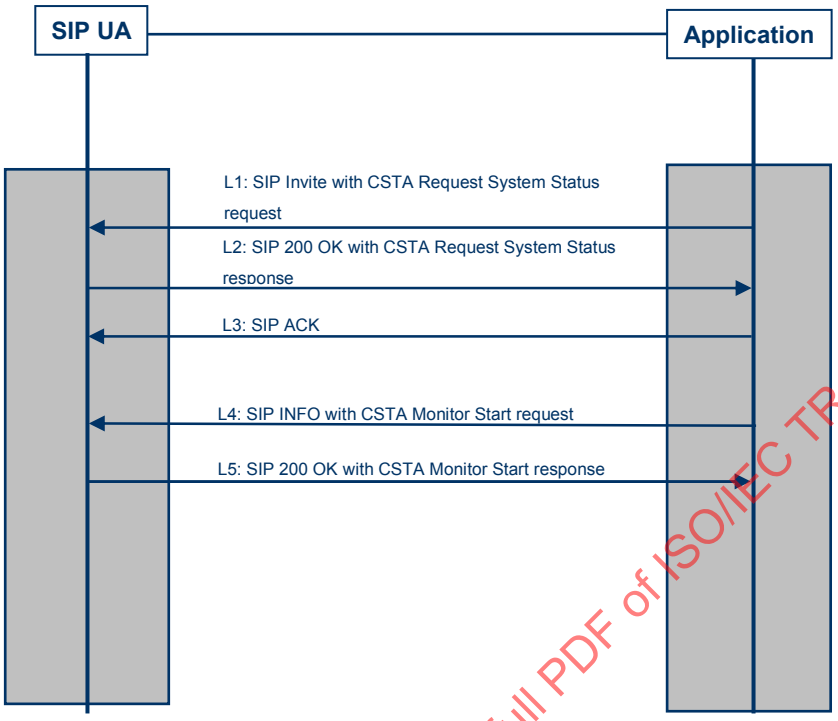
There are two functional components in these examples:

- SIP UA – From a CSTA interface perspective, this is a CSTA “switching function”. The SIP UA application protocol is ECMA-323 over SIP. The CSTA deviceID of the SIP phone is represented with a contact URI of sip:ua1@123.123.123.123.

- Application – From a CSTA interface perspective this is a “computing function”. This component typically includes a client application. The application is also a SIP UA that is represented with a contact URI of sip:app1@pc33.domain.com.

16.1.1 Creating an Application Session, Establishing a Monitor for a SIP Phone

The following figure shows how an application session is created and how a CSTA monitor is started.



The first thing that an application does is establish a CSTA application session with the SIP UA. This is accomplished by sending a SIP INVITE with a CSTA Request System Status service request provided in the body of the INVITE method. The content type is application/csta+xml. The content disposition header specifies that the UA must support the application/csta+xml content type.

In this example, the application sends the INVITE to the user's published address, or Address of Record (AoR) URI sip:tom@domain.com.

Some of the CSTA relevant SIP headers are included in the example below. Refer to IETF RFC 3261 for a complete description of the format of the SIP INVITE and associated headers.

```
INVITE sip:tom@domain.com SIP/2.0
Via:
Max-Forwards:
To: <sip:tom@domain.com>
From: <sip:app1@domain.com>; tag=abc
Call-id:
CSeq: 1 INVITE
Contact: <sip:app1@pc33.domain.com>
Content-Type: application/csta+xml
Content-Disposition: signal; handling=required
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
<RequestSystemStatus xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
```

The UA accepts the Invite and detects (through the Content-Disposition and Content-Type headers) that this is a special Invite to establish a CSTA application session. (This UA supports this MIME type – otherwise it returns a 415 (Unsupported Media Type) response message to the INVITE.)

After the CSTA application association is established, the UA sends a SIP 200 OK to the application as shown in line 2. The 200 OK message includes the CSTA Request System Status response that provides a “normal” system status indicating that a CSTA application session has been established. Note that other system status values could also be provided – in these cases an application session is still considered established but, depending upon the system status value, certain CSTA features may not yet be available.

In this example, the 200 OK includes a contact address URI of sip:ua1@123.123.123.123 that is used to address this UA in subsequent SIP methods such as INFO.

```
SIP/2.0 200 OK
Via:
To: <sip:tom@domain.com>; tag=def
From: <sip:app1@domain.com>; tag=abc
Call-id= xxx
CSeq: 1 INVITE
Contact: <sip:ua1@123.123.123.123>
Content-Type: application/csta+xml
Content-Disposition: signal; handling=required
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
<RequestSystemStatusResponse xmlns="http://www.ecma-international.org/standards/ecma-
323/csta/ed3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <systemStatus>normal</systemStatus>
</RequestSystemStatusResponse>
```

The application responds with an ACK to establish the SIP session for transporting CSTA service requests and responses as shown in line 3.

The app wants to be informed when an incoming call arrives at the UA so it starts a CSTA Monitor on the UA that it is interested in observing (sip:ua1@123.123.123.123). The ECMA-323 Monitor Start service is encapsulated in a SIP INFO method as shown in line 4.

Note that the SIP Request URI header specifies the address returned in the Contact header from the INVITE 200 OK response.

```
INFO sip:ua1@123.123.123.123 SIP/2.0
.
.
Content-Type: application/csta+xml
Content-Disposition: signal; handling=required
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
<MonitorStart xmlns=“http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <monitorObject>
      <deviceObject>sip:ua1@123.123.123.123</deviceObject>
   </monitorObject>
</MonitorStart>
```

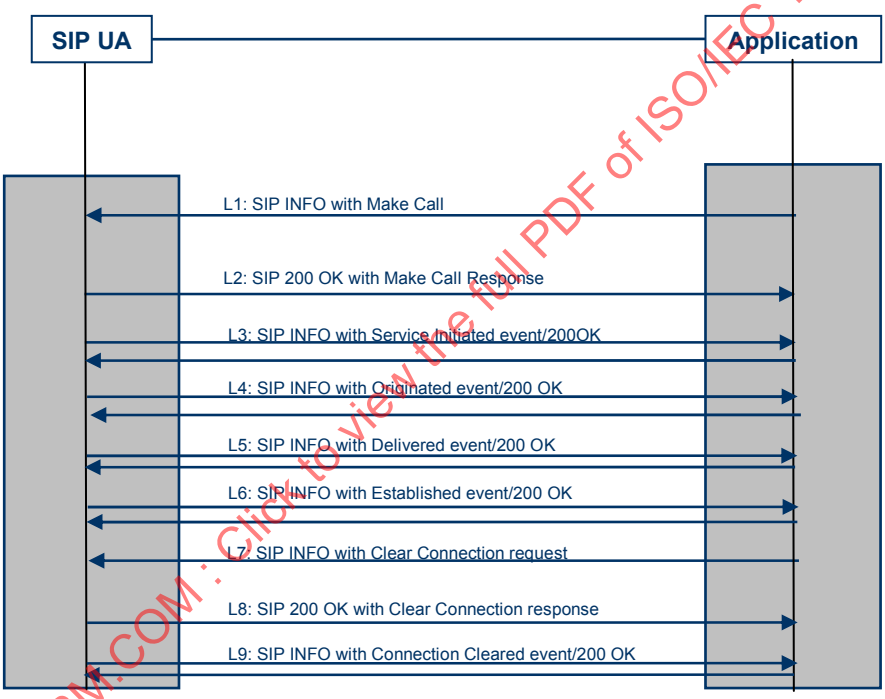
The UA notifies the application that the monitor request was successful by sending a Monitor Start positive response in a 200 OK as shown in line 5.

```
SIP/2.0 200 OK
.
.
Content-Type: application/csta+xml
Content-Disposition: signal; handling=required
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
<MonitorStartResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3 ">
   <monitorCrossRefID>1</monitorCrossRefID>
</MonitorStartResponse>
```

16.1.2 Creating a Call from a SIP UA, Clearing a Call at a SIP UA

The following figure shows an example of a call origination from a SIP UA. The application issues a CSTA Make Call to originate the call. A CSTA monitor has already been established at the originating device.



The application wants to originate a call between SIP UA sip:ua1@123.123.123.123 and SIP UA sip:alice@domain.com. The application wants the originating SIP UA sip:ua1@123.123.123.123 to be auto-originated (not prompted) before sip:alice@domain.com is called. A SIP INFO with a CSTA Make Call service request to achieve this is shown on line 1.

```
INFO sip:ua1@123.123.123.123 SIP/2.0
.
.
Content-Type: application/csta+xml
Content-Disposition: signal; handling=required
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
<MakeCall xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
```

```
    <callingDevice>sip:ua1@123.123.123.123</callingDevice>
    <calledDirectoryNumber>sip:alice@domain.com</calledDirectoryNumber>
    <autoOriginate>doNotPrompt</autoOriginate>
</MakeCall>
```

The UA accepts the INFO with Make Call and sends a Make Call response to the application in a SIP 200 OK as shown in line 2.

```
SIP/2.0 200 OK
.
.
Content-Type: application/csta+xml
Content-Disposition: signal; handling=required
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
<MakeCallResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
    <callingDevice>
        <callID>888</callID>
        <deviceID>sip:ua1@123.123.123.123</deviceID>
    </callingDevice>
</MakeCallResponse>
```

As a result of the call control activity generated as a result of the Make Call service, the SIP UA sends a sequence of ECMA-323 events representing the progress of the call. Each ECMA-323 event is encapsulated in a SIP INFO method.

Note that the following event is ONLY sent if the UA cannot auto-originate the request. If it can auto-originate the request (not prompt the originating device to go off hook) then this event would NOT be sent.

```
INFO sip:app1@pc33.domain.com SIP2/0
.
.
Content-Type: application/csta+xml
Content-Disposition: signal; handling=required
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
<ServiceInitiatedEvent xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
    <monitorCrossRefID>1</monitorCrossRefID>
    <initiatedConnection>
        <callID>888</callID>
        <deviceID>sip:ua1@123.123.123.123</deviceID>
    </initiatedConnection>
    <initiatingDevice>
        <deviceIdentifier>sip:ua1@123.123.123.123</deviceIdentifier>
    </initiatingDevice>
    <localConnectionInfo>initiated</localConnectionInfo>
    <cause>makeCall</cause>
</ServiceInitiatedEvent>
```

The next event indicates that the UA sip:ua1@123.123.123.123 has gone off hook and the call has been originated as shown in line 4.

```
INFO sip:app1@pc33.domain.com SIP2/0
.
.
Content-Type: application/csta+xml
```

```
Content-Disposition: signal; handling=required
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
<OriginatedEvent xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <monitorCrossRefID>1</monitorCrossRefID>
   <originatedConnection>
      <callID>888</callID>
      <deviceID>sip:ua1@123.123.123.123</deviceID>
   </originatedConnection>
   <callingDevice>
      <deviceIdentifier>sip:ua1@123.123.123.123</deviceIdentifier>
   </callingDevice>
   <calledDevice>
      <deviceIdentifier>sip:alice@domain.com</deviceIdentifier>
   </calledDevice>
   <localConnectionInfo>connected</localConnectionInfo>
   <cause>normal</cause>
</OriginatedEvent>
```

The next event indicates that the called SIP UA sip:alice@domain.com is alerting as shown in line 5.

```
INFO sip:app1@pc33.domain.com SIP2/0
.
.
Content-Type: application/csta+xml
Content-Disposition: signal; handling=required
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
<DeliveredEvent xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <monitorCrossRefID>1</monitorCrossRefID>
   <connection>
      <callID>888</callID>
      <deviceID>sip:alice@domain.com</deviceID>
   </connection>
   <alertingDevice>
      <deviceIdentifier>sip:alice@domain.com</deviceIdentifier>
   </alertingDevice>
   <callingDevice>
      <deviceIdentifier>sip:ua1@123.123.123.123</deviceIdentifier>
   </callingDevice>
   <calledDevice>
      <deviceIdentifier>sip:alice@domain.com</deviceIdentifier>
   </calledDevice>
   <lastRedirectionDevice>
      <notRequired/>
   </lastRedirectionDevice>
   <localConnectionInfo>connected</localConnectionInfo>
   <cause>normal</cause>
</DeliveredEvent>
```

The next event indicates that the called SIP phone sip:alice@domain.com has answered the call as shown in line 6.

```
INFO sip:app1@pc33.domain.com SIP2/0
.
.
Content-Type: application/csta+xml
Content-Disposition: signal; handling=required
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
<EstablishedEvent xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <monitorCrossRefID>1</monitorCrossRefID>
   <establishedConnection>
      <callID>888</callID>
      <deviceID>sip:alice@domain.com </deviceID>
   </establishedConnection>
   <answeringDevice>
      <deviceIdentifier>sip:alice@domain.com</deviceIdentifier>
   </answeringDevice>
   <callingDevice>
      <deviceIdentifier>sip:ua1@123.123.123.123</deviceIdentifier>
   </callingDevice>
   <calledDevice>
      <deviceIdentifier>sip:alice@domain.com</deviceIdentifier>
   </calledDevice>
   <lastRedirectionDevice>
      <notRequired/>
   </lastRedirectionDevice>
   <localConnectionInfo>connected</localConnectionInfo>
   <cause>normal</cause>
</EstablishedEvent>
```

Eventually the application wants to clear the call at the SIP UA sip:ua1@123.123.123.123. It sends a CSTA Clear Connection service request to the phone in a SIP INFO as shown in line 7. The connectionID in the CSTA Clear Connection request was obtained from either the Make Call response or one of the above events.

```
INFO sip:ua1@123.123.123.123 SIP/2.0
.
.
Content-Type: application/csta+xml
Content-Disposition: signal; handling=required
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
<ClearConnection xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <connectionToBeCleared>
      <callID>888</callID>
      <deviceID>sip:ua1@123.123.123.123</deviceID>
   </connectionToBeCleared>
</ClearConnection>
```

The SIP UA sends the CSTA Clear Connection response in a 200 OK as shown in line 8.

```
SIP 200 OK
.
.
Content-Type: application/csta+xml
Content-Disposition: signal; handling=required
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
    <ClearConnectionResponse xmlns="http://www.ecma-international.org/standards/ecma-
323/csta/ed3"/>
```

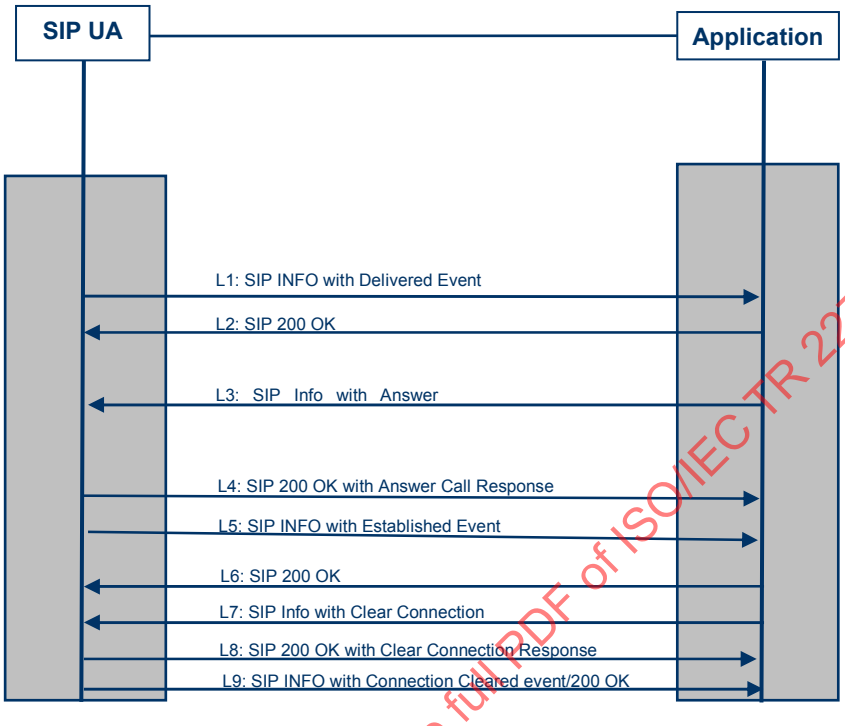
As a result of the Clear Connection service, the call is cleared at sip:tom@domain.com and the SIP UA sends a Connection Cleared encapsulated message to the application as shown in line 9.

```
INFO sip:app1@pc33.domain.com SIP2/0
.
.
Content-Type: application/csta+xml
Content-Disposition: signal; handling=required
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
<ConnectionClearedEvent xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <monitorCrossRefID>1</monitorCrossRefID>
   <droppedConnection>
      <callID>888</callID>
      <deviceID>sip:ua1@123.123.123.123</deviceID>
   </droppedConnection>
   <releasingDevice>
      <deviceIdentifier>sip:ua1@123.123.123.123</deviceIdentifier>
   </releasingDevice>
   <localConnectionInfo>null</localConnectionInfo>
   <cause>normal</cause>
</ConnectionClearedEvent>
```

16.1.3 Answering and Clearing an Incoming Call at a UA

The following example shows an incoming call to a monitored SIP UA. The application detects the incoming call via the CSTA Delivered event and answers the call using the CSTA Answer Call service.



An application has placed a monitor on the SIP UA sip:ua1@123.123.123.123. When an incoming call arrives at this device from SIP phone sip:alice@domain.com, the SIP UA sends a CSTA Delivered event encapsulated in a SIP INFO method to the application as shown in line 1. The application saves the connectionID of the alerting call for use in subsequent CSTA services.

```
INFO sip:app1@pc33.domain.com SIP2/0
.
.
Content-Type: application/csta+xml
Content-Disposition: signal; handling=required
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
<DeliveredEvent xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
   <monitorCrossRefID>1</monitorCrossRefID>
   <connection>
      <callID>999</callID>
      <deviceID>sip:ua1@123.123.123.123</deviceID>
   </connection>
   <alertingDevice>
      <deviceIdentifier>sip:ua1@123.123.123.123</deviceIdentifier>
   </alertingDevice>
   <callingDevice>
      <deviceIdentifier>sip:alice@domain.com</deviceIdentifier>
   </callingDevice>
   <calledDevice>
      <deviceIdentifier>sip:ua1@123.123.123.123</deviceIdentifier>
```

```
    </calledDevice>
    <lastRedirectionDevice><notRequired/></lastRedirectionDevice>
    <localConnectionInfo>alerting</localConnectionInfo>
    <cause>normal</cause>
</DeliveredEvent>
```

The application responds to the SIP INFO with a SIP 200 OK as shown in line 2.

The application uses the CSTA Answer Call service to answer the alerting call at the SIP phone sip:tom@domain.com without requiring manual intervention. The connectionID in the Answer Call service is obtained from the prior Delivered event. The CSTA Answer Call encapsulated in the SIP INFO is shown in line 3.

```
INFO sip:ua1@123.123.123.123 SIP/2.0
.
.
Content-Type: application/csta+xml
Content-Disposition: signal; handling=required
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
<AnswerCall xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
    <callToBeAnswered>
        <callID>999</callID>
        <deviceID>sip:ua1@123.123.123.123</deviceID>
    </callToBeAnswered>
</AnswerCall>
```

The UA sends an Answer Call response in a 200 OK as shown in line 4.

```
SIP 200 OK
.
.
Content-Type: application/csta+xml
Content-Disposition: signal; handling=required
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
<AnswerCallResponse xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3"/>
```

The alerting call at the SIP phone sip:ua1@123.123.123.123 is answered as a result of the CSTA Answer Call service resulting in a CSTA Established encapsulated in a SIP INFO method as shown in line 9.

```
INFO sip:app1@pc33.domain.com SIP2/0
.
.
Content-Type: application/csta+xml
Content-Disposition: signal; handling=required
Content-Length: nnn

<?xml version="1.0" encoding="UTF-8"?>
<EstablishedEvent xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
    <monitorCrossRefID>1</monitorCrossRefID>
    <establishedConnection>
        <callID>999</callID>
        <deviceID>sip:ua1@123.123.123.123</deviceID>
    </establishedConnection>
```

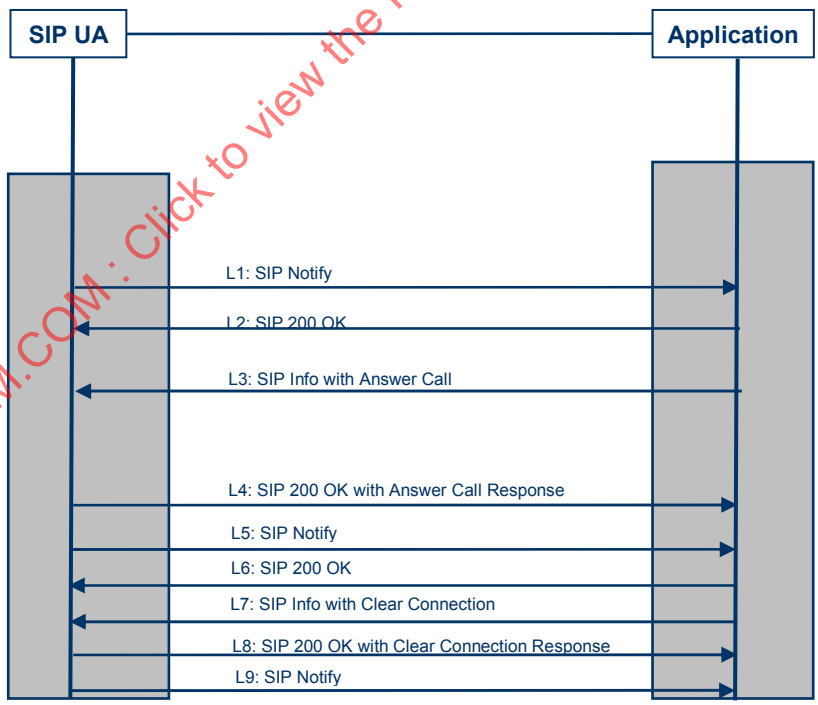
```
    <answeringDevice>
        <deviceIdentifier>sip:ua1@123.123.123.123</deviceIdentifier>
    </answeringDevice>
    <callingDevice>
        <deviceIdentifier>sip:alice@domain.com</deviceIdentifier>
    </callingDevice>
    <calledDevice>
        <deviceIdentifier>sip:ua1@123.123.123.123</deviceIdentifier>
    </calledDevice>
    <lastRedirectionDevice>
        <notRequired/>
    </lastRedirectionDevice>
    <localConnectionInfo>connected</localConnectionInfo>
    <cause>normal</cause>
</EstablishedEvent>
```

Lines 5 and 6 show the application clearing the call as in the previous example.

16.1.4 Answering an Incoming Call at a UA (no CSTA monitor or CSTA events)

These examples show how CSTA service can be used without a CSTA monitor. For example, in the Minimal uaCSTA Call Control profile, an application could use a SIP event package to obtain the information on a call that can be used for a CSTA connectionID in a CSTA service request.

The following example shows an incoming call to a SIP UA. The application detects the incoming call and the state of the SIP dialog through a SIP event package. It uses the CSTA Answer Call service to answer the call.



When an incoming call arrives at this UA, the UA sends a SIP Notify to the application. This is the result of a previous subscription to a SIP event package that provides dialog state information.