



ecma

INTERNATIONAL

What is CSTA?

CSTA Overview

Started by Tom Miller (Siemens),
updated by Ecma/TC32-TG11,
December 2005.

CSTA History

CSTA Standards Suite

CSTA Features

ECMA-323 (CSTA XML)

Call Control Details

Voice Browsers & ECMA-323

Current CSTA Standardization Activities

Abstraction Layer for telecommunication applications:

- *Independent of underlying signaling protocols*
 - **H.323, SIP, Analog, T1, ISDN, etc.**
- *Independent of devices*
 - **intelligent endpoints, low-function/stimulus devices, SIP Signaling models - 3PCC vs. Peer/Peer**

Operates equally well in different environments:

- *3rd party call control*
- *1st party call control*

Basic call model standardized in 1992 – continually refined and enhanced based upon significant industry implementation experiences, new protocols, etc.

Design goal to enhance application portability across CSTA implementations:

- *Specifies normalized call model and behavior*
- *Complete functional definition of each service*
- *Specific conformance criteria*

Phase I

- **published June '92**
- **40 features, 66 pages (Services Specification)**
- **focus on call control**

Phase II

- **published Dec. '94**
- **77 features, 145 pages (Services Specification)**
- **I/O & Voice Unit Services, more call control services**

Phase III - CSTA Phase II Features & versit CTI Technology

- **Published Dec. '98**
- **136 features, 650 pages (Services Specification)**
- **Published as an ISO Standard July 2000**
- **Published CSTA XML (ECMA-323) June 2004**
- **Published "Using CSTA with Voice Browsers" (TR/85) December 2002**
- **Published CSTA WSDL (ECMA-348) June 2004**
- **Published Object Model TR/88 June 2004**
- **Published "Using CSTA for SIP Phone User Agents (uaCSTA)" TR/87 June 2004**

Document Title	Ecma Publication	ISO/IEC Publication	ETSI Publication
Services for CSTA Phase III	<u>ECMA-269</u>	<u>ISO/IEC 18051</u>	<u>ETSI TS 102 173</u>
ASN.1 Protocol for CSTA Phase III	<u>ECMA-285</u>	<u>ISO/IEC 18052</u>	
XML Protocol for CSTA Phase III	<u>ECMA-323</u>	<u>ISO/IEC 18056</u>	<u>ETSI TS 102 174</u>
Web Services Description Language (WSDL) for CSTA Phase III	<u>ECMA-348</u>		
Definitions & Terms for CSTA Ph. III	<u>ECMA TR/72</u>	<u>ISO/IEC TR 18053</u>	
Migrating to CSTA Phase III	<u>ECMA TR/80</u>		
Scenarios for CSTA Phase III	<u>ECMA TR/82</u>		
Using Ecma-323 (CSTA XML) in a Voice Browser Environment	<u>ECMA TR/85</u>	<u>ISO/IEC TR 18057</u>	<u>ETSI TR 102 171</u>
Using CSTA for SIP Phone User Agents (uaCSTA)	<u>ECMA TR/87</u>	<u>ISO/IEC TR 22767</u>	<u>ETSI TR 102 348</u>
Designing an Object Model for ECMA-269 (CSTA)	<u>ECMA TR/88</u>		

Document Title	Ecma Publication	ISO/IEC Publication	ETSI Publication
Application Session Services (used in ECMA-269)	<u>ECMA-354</u>	<u>ISO/IEC 22534</u>	<u>ETSI TS 102 344</u>
WS-Session – (WSDL for ECMA-354)	<u>ECMA-366</u>	<u>ISO/IEC 25437</u>	<u>ETSI TS 102 440</u>
Session Management, Event Notification, and Computing Function Services - Amendments for ECMA-348	<u>ECMA TR/90</u>		

CSTA Features

- 26 Call Control features (making call, answering call, etc.)
- 6 Call Associated features (sending user data, etc.)
- 19 Logical Device features (do not disturb, forwarding, etc.)
- 23 Physical Device features (writing to device display, etc.)
- 5 Capability Exchange features (feature discovery, etc.)
- 4 Snapshot features (query existing calls at a device, etc.)
- 3 Monitor features (subscribing to event reports, etc.)
- 17 Voice Services (for Listener, DTMF, Prompt and message resources)

Other: Routing services, Media Attachment services, Maintenance services, Data Collection services, Accounting services, etc.

Implementation does not need to support all of these features to conform to CSTA! (See Slide on Profiles)

Exhaustive set of standardized call control features:

- **Services:** *Accept, Alternate, Answer, Call Back, Call Back Message, Camp On, Clear Call, Clear Connection, Conference, Consultation, Deflect, Dial Digits, Directed Pick up, Group Pick Up, Hold, Intrude, Join, Make Call, Make Predictive Call, Park, Reconnect, Retrieve, Send Message, Single Step Conference, Single Step Transfer, Transfer.*
- **Events:** *Bridged, Call Cleared, Conferenced, Connection Cleared, Delivered, Digits Dialed, Diverted, Established, Failed, Held, Network Capabilities Changed, Network Reached, Offered, Originated, Queued, Retrieved, Service Initiated, Transferred.*

(features included in Basic Telephony profile are underlined)

Each feature includes a complete operational model on how the feature works, from an application perspective.

- *Before/After conditions*
- *State transitions*
- *Event flow*
- *Textual behavior description*

Alerting – Indicates an incoming call at an endpoint. 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 represent that an outgoing call attempt that encountered a busy endpoint.

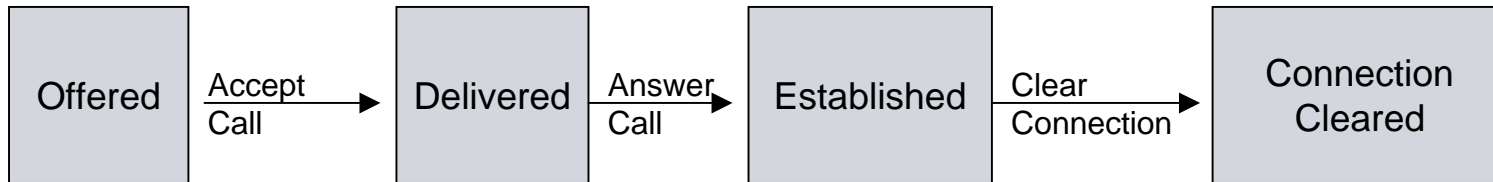
Held – Indicates that an endpoint is no longer actively participating in a call. For implementations that support multiple calls per endpoint (i.e. line), a connection could be Held while the line is used to place another call (consultation transfer on an analog line, for example).

Initiated – A transient state, usually indicating that the endpoint is initiating a service (e.g. dialtone) or the device is being prompted to go offhook.

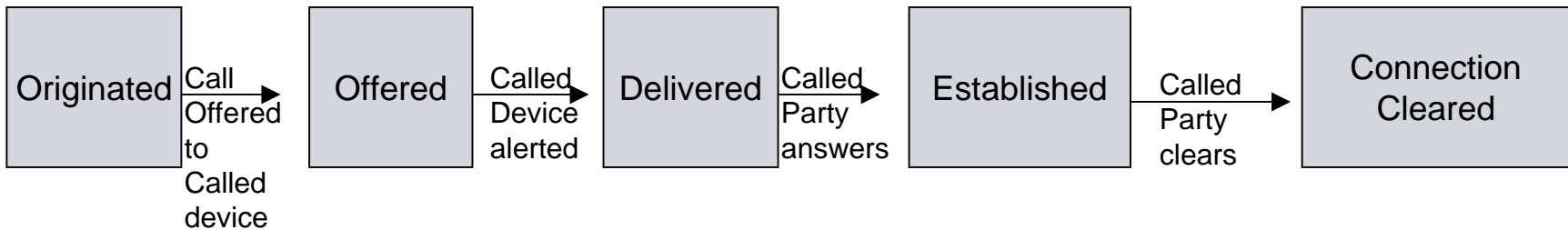
Null – There is no relationship between the call and the endpoint.

Queued – Indicates that the call is temporarily suspended at a device (e.g. call has been parked, camped on).

typical call events for an incoming call



typical call events for an outgoing call (Make Call)



Standard is extensible:

- *Additional features can be added to the standard via new editions of the standard*
- *New Parameters, new values to existing parameters can be added via new editions of the standard*
- *Backward compatibility must be maintained in any future editions*
- *Implementations can add proprietary features using built in CSTA extension mechanisms*

CSTA Includes a set of Profiles:

- *Specifies minimal set of features required to conform to CSTA*
- *At least one profile must be supported by an CSTA communication provider*
- *Applications can be developed with feature set in mind*

Existing Profiles:

- *Basic Telephony profile*
- *Routing profile*
- *3 Voice Browser profiles (new in ECMA-269, 5th ed.)*
- *6 SIP phone (uaCSTA) profiles (new in ECMA-269, 6th ed.)*

XML Encoding for CSTA feature set

- *Set of W3C XML Schemas*
- *One schema per service/event*

Contains XML encoding for all CSTA features standardized in ECMA-269

CSTA XML facilitates use of CSTA features by Internet developers – when combined with Scripting languages such as [ECMAScript](#), it becomes very easy to use the CSTA XML interface directly

```
<?xml version="1.0" encoding="UTF-8"?>  
<MonitorStart xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">  
  <monitorObject>  
    <deviceObject>22343</deviceObject>  
  </monitorObject>  
</MonitorStart>
```

Answering an Alerting Call

```
<?xml version="1.0" encoding="UTF-8"?>
<AnswerCall xmlns="http://www.ecma-
international.org/standards/ecma-233/csta/ed3">
  <callToBeAnswered>
    <callID>1</callID>
    <deviceID>22343</deviceID>
  </callToBeAnswered>
</AnswerCall>
```


Clearing a connection

```
<?xml version="1.0" encoding="UTF-8"?>
<ClearConnection xmlns="http://www.ecma-
international.org/standards/ecma-323/csta/ed3">
  <connectionToBeCleared>
    <callID>1</callID>
    <deviceID>22343</deviceID>
  </connectionToBeCleared>
</ClearConnection>
```

Notification of Incoming Call

Rich content – application “picks” info it needs

```
<DeliveredEvent xmlns="http://www.ecma-international.org/standards/ecma-323/csta/ed3">
  <monitorCrossRefID>99</monitorCrossRefID>
  <connection>
    <callID>1</callID>
    <deviceID>22343</deviceID>
  </connection>
  <alertingDevice><deviceIdentifier>22343</deviceIdentifier></alertingDevice>
  <callingDevice><deviceIdentifier>14085551212</deviceIdentifier></callingDevice>
  <calledDevice><deviceIdentifier>22343</deviceIdentifier></calledDevice>
  <lastRedirectionDevice><notRequired/></lastRedirectionDevice>
  <localConnectionInfo>alerting</localConnectionInfo>
  <cause>newCall</cause>
  <networkCallingDevice><deviceIdentifier>14085551212</deviceIdentifier>
</networkCallingDevice>
  <networkCalledDevice><deviceIdentifier>18001234567</deviceIdentifier>
</networkCalledDevice>
  <associatedCallingDevice><deviceIdentifier>023</deviceIdentifier>
</associatedCallingDevice>
</DeliveredEvent>
```

CSTA XML ideally suited for VB platforms that support a messaging interface w/ asynchronous events (such as SALT smex)

ECMA-323 chosen as the recommended call control messaging interface for SALT

- *Alternative to SALT CC Object Model*
- *Used with the SALT platform messaging (smex)*
- *Easy to create & access ECMA-323 XML messages via ECMAScript/DOM*
- *Examples of ECMA-323, ECMAScript, SALT in SALT 1.0*

CSTA support for Voice Browsers:

- *Added profiles optimized for Voice Browser applications.*
- *Published a Technical Report (TR/85) that shows how ECMA-323 can be used in a Voice Browser environment.*

Support for non-voice media interactions (IM, Email, Chat)

- *CSTA call model applicable to non-voice media (Email, Chat, IM, etc.)*
- *CSTA "call" and "connection" objects are media independent*
- *chat can be modeled as an interactive "text call"*
- *Email can be modeled as an non-interactive "text call"*
- *Additional parameters for message information, subject of call, priority, sensitivity of calls, etc.*

Enhancements to improve SIP support

- *Features to improve control of media (connection information), support of SIP 3PCC, etc.*

- *Interactive voice features to support advanced speech applications*
- *Service to obtain a list of CSTA features*
- *Profiles for SIP user agents*
- *New methods for establishing CSTA applications sessions*
- *Internet URI device format*
- *Device ID character support for Intl. Numbers*
- *Removed size constraints for parameters*
- *Media Class types to support IM, SMS and MMS*
- *User model to support monitoring a User versus a Device*

Speech service enhancements to CSTA

- *For speech recognition/verification*
- *For speaker recognition/verification*
- *For text to speech synthesis*
- *For distributed speech services using:*
 - **ECMA-323 over SIP or TCP/IP**
 - **ECMA-348 for Web Services**
- *Added speech resources to CSTA:*
 - **Listener, Prompt, Prompt-Queue, DTMF, Message and Generic**
- *Added Interactive Speech Devices to CSTA*
 - **Enables seamless integration of speech and call control**

TR shows how to tunnel CSTA using SIP:

- *Added profiles to ECMA-269 tailored to SIP environments*
- *Enhanced CSTA to better support SIP (e.g. URI addressing)*
- *TR illustrates deployment examples*
- *TR shows how relevant CSTA concepts map to SIP*
- *No changes to SIP, using existing SIP INFO method and registered CSTA MIME types*
- *Enables rich behavior of SIP phones*
- *Augments SIP features (e.g. SIP 3PCC)*

Published TR in June 2004

- *Scope*
 - **Call controls: same as voice browser TR ([ECMA TR/85](#))**
 - **Speech Services: same as SALT 1.0**
- *ECMA-335 (CLI) based design*
 - **Sample code in ECMA-334 (C#)**
- *TR aims to solicit (developer) feedback*
- *Goal*
 - **Use feedback to finalize object model design**
 - **Publish the Object Model Standard in mid 2005**

Application Association:

```
mySF = new CSTA.Provider();
```

Obtain a device:

```
myDevice = mySF.GetDevice(DeviceID(0001));
```

Register an event handler:

```
myDevice.EstablishedEvent +=  
    new EstablishedEventHandler(myHandler);
```

Start Monitoring:

```
myDevice.MonitorStart();
```

Make an outbound call:

```
myConnection = myDevice.MakeCall("5551212");
```

Hangup:

```
myConnection.ClearConnection();
```

Detailed references:

- *Connection model (ECMA-269: 6.5.1)*
- *Monitoring Concepts (ECMA-269: 15)*
- *Snapshot Services (ECMA-269: 16)*
- *Summary of Parameter Types (elements) used in ECMA-323 messages (ECMA-269: 12.2)*
- *Call Control Services Walkthrough (ECMA-269: 17.1)*
- *Call Control Events Walkthrough (ECMA-269: 17.2)*

Technical Report uaCSTA

- *Process feedback on uaCSTA TR*

Work Item to Enhance CSTA with advanced Conferencing Services

- *To support finer grained of media control for voice paths to enable "whisper, sidebar, split, combine" interactions*

Object Model Standard for CSTA

- *Incorporate lessons learned from the Object Model Technical Report*
- *Expand scope from voice browser to fully cover ECMA-269*

Enhancements to CSTA for NGN, mobility, carrier grade environments

CSTA is an existing (Ecma, ETSI, ISO) Standard with an exhaustive feature set, comprehensive call model

CSTA supports range of application landscapes – from basic 1st party call control to advanced 3rd party call control with same standardized model

CSTA exposes advanced features of a communications platform to applications developers while insulating applications from underlying protocol specifics

CSTA XML facilitates use of call control features by Internet developers – when combined with Scripting languages such as ECMAScript, it becomes easy to program directly to the CSTA XML interface

CSTA XML ideally suited for VB platforms that support a messaging interface w/ asynchronous events (such as SALT smex); supports advanced Interactive Voice applications

CSTA supports voice and non-voice interactions (Email, Chat, IM, etc.) with the same call model.

CSTA complements SIP and enables developers to provide advanced features

CSTA Object Model TR provides a robust and current access method for CSTA