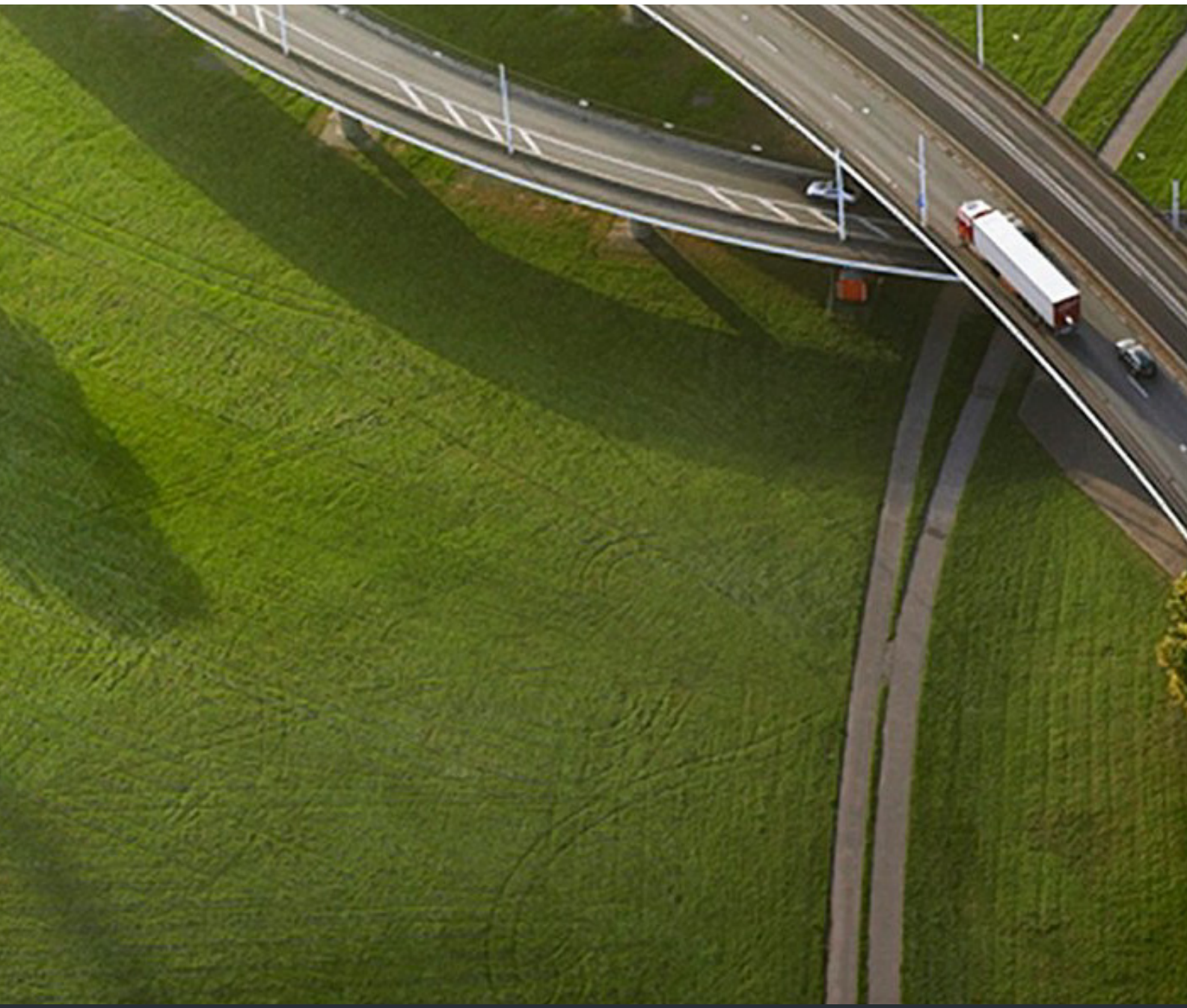
UNIFY Harmonize your enterprise



OpenScape Voice V9
Interface Manual: Volume 4, CSTA Interface

Description

A31003-H8090-T103-01-7618

Our Quality and Environmental Management Systems are implemented according to the requirements of the ISO9001 and ISO14001 standards and are certified by an external certification company.

Copyright © Unify Software and Solutions GmbH & Co. KG 01/2016
Mies-van-der-Rohe-Str. 6, 80807 Munich/Germany

All rights reserved.

Reference No.: A31003-H8090-T103-01-7618

The information provided in this document contains merely general descriptions or characteristics of performance which in case of actual use do not always apply as described or which may change as a result of further development of the products. An obligation to provide the respective characteristics shall only exist if expressly agreed in the terms of contract.

Availability and technical specifications are subject to change without notice.

Unify, OpenScape, OpenStage and HiPath are registered trademarks of Unify Software and Solutions GmbH & Co. KG. All other company, brand, product and service names are trademarks or registered trademarks of their respective holders.

UNIFY Harmonize your enterprise

unify.com

Contents

History of Changes 10

1 Introduction to this Guide 11
1.1 Scope 11
1.2 Audience 11
1.3 What's New in OpenScape Voice V9 V9 / V8R1 12
1.3.1 Fallback to 3PCC Make Call 12
1.3.2 One-way speaker call broadcast 12
1.3.3 Parallel - Simultaneous Alerting Model 14
1.3.4 CSTA change for hold/retrieve functionality 14
1.3.5 Provide accumulated busy state for MLHGs / MLHG Pilot 14
1.3.6 CSTA and OSMO/WebRTC Enhancements for Callback on Busy and No Reply 15
1.3.7 Centralized Call Log 15
1.3.8 Do Not Disturb (DND) with Snooze Option 16
1.3.9 CSTASM to extend the length of the appSessionID and block Reset of an active session 16
1.3.10 Introduce a noReOffer value to the noReOffer privateData flag to prevent delay in call processing in certain scenario 16
1.3.11 OpenScape_Voice V8 0 Branding 17
1.3.12 Enable modification of CSTA Application Interface Options, with an appropriate warning, when a CSTA application status is connected 17
1.3.13 ODC WE, web client shown no green phone icon to accept the incoming video call when 3 incoming video calls left no answer 17
1.3.14 WebClient user cannot leave a callback against another party after not accepting a previous callback return call 17
1.3.15 CSTA-SM need to implement a random number for applicationSessionId and add a check/log entry for bad appSession records to the audit 18
1.3.16 General Corrections and Updates 18
1.3.16.1 CSTA over HTTP/SOAP Interface 18
1.3.16.2 Application Interface Management 18
1.4 References 19
1.4.1 External CSTA Documentation 19
1.4.2 Unify Documentation 20

2 CSTA Service Overview 21
2.1 OpenScape Voice CSTA Operational Model 21
2.2 ECMA CSTA Compliance Statement 21
2.2.1 ECMA-323 Namespace 21
2.2.2 ECMA-323 Specified Tags 21
2.3 CSTA Application and OpenScape Voice Configurations 22
2.3.1 Direct Connect Configurations 22
2.3.2 CSTA Transport Considerations 23
2.3.2.1 CSTA over TCP 23
2.3.2.2 CSTA over SIP (for SIMPLE uaCSTA) 23
2.3.2.3 CSTA over HTTP/SOAP 26
2.3.3 CSTA over SIP 27
2.3.4 QoS DSCP/TOS 27
2.3.5 Application Link Security Considerations 29
2.3.5.1 CSTA Application White List 29

2.3.5.2 Transport Security using IPSec . . . 29
2.3.5.3 Transport Security via TLS (not supported) . . . 29
2.4 Supported CSTA Services and Events (by Device Type) . . . 30
2.5 Recommended CSTA Device Type Settings . . . 31
2.6 Common CSTA Parameter Considerations . . . 32
2.6.1 DeviceID Number and Name Presentation Conventions . . . 32
2.6.1.1 CSTA Device Identification Number . . . 32
2.6.1.2 Global Number Format . . . 38
2.6.1.3 Device Number (DeN) Format . . . 38
2.6.1.4 Number and Name Display Including Privacy . . . 39
2.6.1.5 Multibyte Support for Name (NM) Field in UTF-8 Encoding . . . 43
2.6.2 Negative Service Response . . . 43
2.6.3 Dynamic Feature Presentation (using servicesPermitted) . . . 47
2.6.3.1 General . . . 47
2.6.3.2 Device Type Considerations . . . 50
2.6.3.3 Call State Considerations . . . 51
2.6.3.4 Situational Considerations . . . 52
2.6.3.5 Static Condition Considerations . . . 53
2.6.3.6 Service Permitted Usage Notes . . . 53
2.7 OpenScape Voice Feature Interaction with CSTA and ONS . . . 54

3 Administration and Management Considerations for CSTA . . . 55
3.1 Application Connectivity . . . 55
3.1.1 OpenScape Voice—CSTA over TCP Connections . . . 55
3.1.2 CSTA Application Connectivity to OpenScape Voice V9 Duplex Cluster . . . 56
3.1.2.1 Co-Located Node Cluster Failover . . . 56
3.1.2.2 Geographically Separated Node Cluster Failover . . . 58
3.1.2.3 Effects on CSTA Call Data on Node Failover . . . 59
3.1.2.4 Connection Failure due to Network Problems . . . 59
3.2 Subscriber Management Configuration (OpenScape Voice Assistant) or Expert Mode using CLI . . . 60
3.2.1 BGL Services Affecting CSTA Services . . . 60
3.2.2 BGL with Registering or Non-Registering SIP Device . . . 60
3.2.2.1 Type1 and CSTA over SIP Device Type configuration . . . 61
3.2.3 MLHG Pilot (Group Device / ACD) and Agent BGLs . . . 62
3.2.4 One Number Services (ONS) / ONS Subscribers . . . 63
3.2.5 SIP Multiple Contact Registration and Feature Interaction . . . 63
3.2.6 **Shared Multiple Appearance (Keyset Operation)** . . . 64
3.2.7 Multiple Contacts (SIP Forking) . . . 65
3.2.8 Global Number Format (GNF) Replacement - RTP Parameter . . . 65
3.3 Failure and Restart Considerations . . . 66
3.3.1 Simplex OpenScape Voice System . . . 66
3.3.2 Duplex OpenScape Voice System (co-located) . . . 66
3.3.3 Duplex OpenScape Voice System (geo/network separated) . . . 66
3.3.3.1 CSTA Application using Cold Stand-By Operation . . . 66
3.3.3.2 CSTA Application using Hot Stand by Operation . . . 66
3.4 ECMA 354 Application Session Services . . . 67
3.5 Call Admission Control (CAC) Subscriber Re-routing . . . 67
3.6 CSTA Application Management . . . 67

4 OpenScape Voice CSTA Service Description . . . 69
4.1 Capabilities Exchange . . . 69
4.1.1 Services . . . 69
4.1.1.1 Get CSTA Features . . . 69

4.1.1.2 Get Logical Device Information . . . 70
4.1.1.3 Get Switching Function Capabilities . . . 74
4.1.1.4 Get Physical Device Information . . . 78
4.1.1.5 Get Switching Function Devices . . . 78
4.1.1.6 Switching Function Devices . . . 79
4.2 System Services . . . 81
4.2.1 Registration Services . . . 81
4.2.1.1 Change System Status Filter . . . 81
4.2.1.2 System Register . . . 81
4.2.1.3 System Status Register Abort . . . 82
4.2.1.4 System Status Register Cancel . . . 82
4.2.2 Services . . . 83
4.2.2.1 Request System Status . . . 83
4.2.2.2 System Status . . . 84
4.2.2.3 Switching Function Capabilities Changed . . . 88
4.2.2.4 Switching Function Device Changed . . . 88
4.3 Monitoring Service . . . 89
4.3.1 Services . . . 89
4.3.1.1 Change Monitor Filter . . . 89
4.3.1.2 Monitor Start . . . 90
4.3.1.3 Monitor Stop . . . 93
4.4 Snapshot Services . . . 94
4.4.1 Services . . . 94
4.4.1.1 Snapshot Call . . . 94
4.4.1.2 Snapshot Device . . . 96
4.4.1.3 Snapshot CallData . . . 98
4.4.1.4 Snapshot DeviceData . . . 98
4.5 Application Session Services . . . 100
4.5.1 Start Application Session . . . 101
4.5.1.1 Basic Application Session (Monitor Recovery) . . . 103
4.5.1.2 Application Session with Application Heartbeat (Fallback Call Distribution) . . . 104
4.5.2 Stop Application Session . . . 106
4.5.3 Reset Application Session Timer . . . 107
4.5.4 Application Session Terminated . . . 108
4.6 Call Control . . . 109
4.6.1 Services . . . 109
4.6.1.1 Accept Call (ONS Subscribers Only) . . . 109
4.6.1.2 Alternate Call . . . 110
4.6.1.3 Answer Call . . . 111
4.6.1.4 Call Back Call-Related . . . 113
4.6.1.5 Camp on Call . . . 115
4.6.1.6 Clear Call . . . 115
4.6.1.7 Clear Connection . . . 116
4.6.1.8 Conference Call . . . 118
4.6.1.9 Consultation Call . . . 120
4.6.1.10 Deflect Call . . . 124
4.6.1.11 Dial Digits . . . 125
4.6.1.12 Directed Pickup Call . . . 125
4.6.1.13 Group Pickup Call . . . 125
4.6.1.14 Hold Call . . . 126
4.6.1.15 Intrude Call . . . 127
4.6.1.16 Join Call . . . 127

4.6.1.17 Make Call . . . 130
4.6.1.18 Make Predictive Call . . . 134
4.6.1.19 Park Call . . . 135
4.6.1.20 Reconnect Call . . . 136
4.6.1.21 Retrieve Call . . . 137
4.6.1.22 Send Message . . . 137
4.6.1.23 Single Step Conference Call . . . 138
4.6.1.24 Single Step Transfer Call . . . 139
4.6.1.25 Transfer Call . . . 142
4.6.2 Events . . . 144
4.6.2.1 Bridged . . . 144
4.6.2.2 Call Cleared . . . 144
4.6.2.3 Conferenced . . . 144
4.6.2.4 Connection Cleared . . . 147
4.6.2.5 Delivered . . . 150
4.6.2.6 Dialed Digits . . . 153
4.6.2.7 Diverted . . . 154
4.6.2.8 Established . . . 157
4.6.2.9 Failed . . . 160
4.6.2.10 Held . . . 164
4.6.2.11 Network Capabilities Changed . . . 165
4.6.2.12 Network Reached . . . 166
4.6.2.13 Offered . . . 169
4.6.2.14 Originated . . . 174
4.6.2.15 Queued . . . 177
4.6.2.16 Retrieved . . . 181
4.6.2.17 Service Initiated . . . 183
4.6.2.18 Transferred . . . 186
4.7 Call Associated Features . . . 189
4.7.1 Services . . . 189
4.7.1.1 Associate Data . . . 189
4.7.1.2 Cancel Telephone Tones . . . 190
4.7.1.3 Change Connection Information . . . 191
4.7.1.4 Generate Digits . . . 192
4.7.1.5 Generate Telephony Tones . . . 193
4.7.1.6 Send User Information . . . 193
4.7.2 Events . . . 194
4.7.2.1 Call Information . . . 194
4.7.2.2 Charging . . . 196
4.7.2.3 Digits Generated . . . 197
4.7.2.4 Telephony Tones Generated . . . 197
4.7.2.5 Service Completion Failure . . . 197
4.8 Media Attached Services . . . 202
4.9 Routing Services . . . 203
4.9.1 Registration Services . . . 203
4.9.1.1 Route Register . . . 203
4.9.1.2 Route Register Abort . . . 205
4.9.1.3 Route Register Cancel . . . 206
4.9.2 Services . . . 207
4.9.2.1 Re-Route (with Loop Prevention) . . . 207
4.9.2.2 Route End . . . 209
4.9.2.3 Route Reject . . . 211

4.9.2.4 Route Request . . . 212
4.9.2.5 Route Select . . . 215
4.9.2.6 Route Used (not supported) . . . 216
4.10 Physical Device Features . . . 217
4.10.1 Services . . . 217
4.10.1.1 Get Message Waiting Indicator . . . 218
4.10.1.2 Get Microphone Mute . . . 219
4.10.1.3 Set Message Waiting Service . . . 220
4.10.1.4 Get Speaker Volume . . . 222
4.10.1.5 Set Microphone Mute . . . 223
4.10.1.6 Set Speaker Volume . . . 224
4.10.2 Events . . . 225
4.10.2.1 Message Waiting . . . 225
4.11 Logical Device Feature Services . . . 225
4.11.1 Services . . . 225
4.11.1.1 Call Back Non-Call Related . . . 226
4.11.1.2 Cancel Call Back . . . 227
4.11.1.3 Get Agent State . . . 227
4.11.1.4 Set Agent State . . . 229
4.11.1.5 Get Do Not Disturb . . . 231
4.11.1.6 Set Do Not Disturb . . . 233
4.11.1.7 Get Forwarding . . . 235
4.11.1.8 Set Forwarding . . . 237
4.11.2 Events . . . 239
4.11.2.1 Agent Busy . . . 239
4.11.2.2 Agent Not Ready . . . 240
4.11.2.3 Agent Ready . . . 242
4.11.2.4 Agent Working After Call . . . 243
4.11.2.5 Call Back Event . . . 245
4.11.2.6 Do Not Disturb . . . 246
4.11.2.7 Forwarding . . . 248
4.12 Device Maintenance . . . 250
4.12.1 Events . . . 250
4.12.1.1 Back in Service . . . 251
4.12.1.2 Device Capabilities Changed . . . 253
4.12.1.3 Out of Service . . . 254
4.13 I/O Services . . . 255
4.14 Vendor Specific Extensions Service . . . 255
4.15 Data Collection Services . . . 255
4.16 Voice Services and Events . . . 255

5 Internal Service Framework . . . 257
5.1 Centralized Call Log (CCL) Service . . . 257
5.1.1 Private Namespace (non-ECMA namespace) . . . 257
5.1.2 XML Short Tags Definition . . . 258
5.1.3 Centralized Call Log Services and Events . . . 260
5.1.3.1 CSTA Call Control Event Tracking . . . 260
5.1.3.2 Call Log Services Description - Call Log Snapshot (CF ->SF) . . . 260
5.1.3.3 Call Log Services Description - Call Log Delete Entry (CF -> SF) . . . 262
5.1.3.4 Call Log Event Description - Call Log Event (SF -> CF) . . . 263
5.1.3.5 Event Flow Examples . . . 267

A CSTA Call Scenarios 270

B OpenScape Voice privateData Schema 271
B.1 OpenScape Voice Private Data Schema - ECMA 323 and ECMA 354 271
B.2 OpenScape Voice Internal Services Schema 271
B.2.1 Centralized Call Log 271

C Supported CSTA Services and Events by Device Type 272

D Number Plan Considerations 283
D.1 Configuring BGLs with Private Network Numbers Only 284
D.1.1 Office Code 284
D.1.2 Home Directory Numbers 286
D.1.3 Subscribers 286
D.1.4 Dial Plans and Routing 286
D.1.4.1 Direct Routing from Destinations Codes Table to Home DN 286
D.1.4.2 Indirect Routing 288
D.1.5 Number Modification 290
D.1.6 Restrictions 290

E MLHG and CSTA Capabilities 294
E.1 Basic Hunt Group Attributes 295
E.1.1 Hunt Group – Pilot DN 295
E.1.2 Hunting and Distribution types (Hunting Algorithms) 295
E.1.3 Queuing Attributes 297
E.1.3.1 Maximum Queue size 297
E.1.3.2 Configured Queue size 297
E.1.3.3 Queue Priority 297
E.1.3.4 Overflow Destination 297
E.1.3.5 Maximum Time In Queue 297
E.1.3.6 Intercept Announcement 297
E.1.3.7 Queue Position Announcement Interval and Queue Position Announcement 297
E.1.3.8 Maximum Time In Queue 298
E.1.4 Hunting Attributes 299
E.1.4.1 No Answer Advance 299
E.1.4.2 Auto Make Busy 299
E.1.4.3 Post Call Time (e.g., Wrap-up) 299
E.1.4.4 Collect Traffic Measurement Data 299
E.1.5 Night Service Attributes 300
E.1.5.1 Enable Night Service 300
E.1.5.2 Night Directory Number 300
E.1.5.3 Attendant 300
E.1.6 Member / Agent Attributes 300
E.1.6.1 Subscriber ID 300
E.1.6.2 Position 300
E.1.6.3 Queue Priority 300
E.1.6.4 Busy Status 301
E.1.6.5 Busy Stop Hunt 301
E.1.6.6 Can Make Hunt Group Busy 301
E.1.6.7 Can Stop Hunt Group Hunting 301
E.1.6.8 Attendant 301
E.1.7 Call Admission Control and WAN Failure 302
E.1.7.1 Enable Enhanced Subscriber Rerouting 302
E.1.7.2 Enable Basic Subscriber Rerouting 302

E.1.8 Calling Party Display 302
E.1.9 Manual (Application Controlled) Hunting and Backup Call Distribution (BCD). 303
F One Number Service (ONS) 304
List of Abbreviations 305
Index. 308

History of Changes

Issue	Date	Changes
1	Jan 2016	Initial Release for OpenScape Voice V9

1 Introduction to this Guide

1.1 Scope

The OpenScape Voice CSTA Interface Specification defines the CSTA Services that allow functional integration between any CSTA-enabled application (computing function) and OpenScape Voice (switching function).

The information in this specification is based on a suite of ECMA CSTA Standards and Technical Reports (Refer to European Computer Manufacturers Association (ECMA)). The ECMA CSTA specification provides a comprehensive description of the architectural and practical issues involved in applying, implementing, and utilizing CSTA-based CTI applications.

The OpenScape Voice basic CSTA Interface Specification consists of the following parts:

- **PART I - CSTA Service Description** is written in a format similar to ECMA 269 and ECMA 354 and defines specific OpenScape Voice CSTA Service implementation characteristics.
- **PART II - CSTA Event Flows** are written in a format similar to ECMA TR/82 and describes specifics call flows between a CSTA-enabled application and the OpenScape Voice. *These event flows are provided as examples and are subject to change without notice to third-party application providers.*
- **PART III - CSTA Private Data Schema** provides the OpenScape Voice V9 Private Data Schema using ECMA-323 and ECMA 354 guidelines.

Unless otherwise stated, supported OpenScape Voice CSTA services are compliant with the operational requirements described in the ECMA standards and technical reports listed in Section 1.4, “References”.

1.2 Audience

It is assumed that the reader of this specification has a basic working knowledge of ECMA CSTA standards and technical reports. The ECMA International web site provides an excellent CSTA highlights presentation (Refer to European Computer Manufacturers Association (ECMA))

1.3 What's New in OpenScape Voice V9 V9 / V8R1



The following describes new features for OpenScape Voice V9 V9 and backward compatibility statement for the CSTA Interface:

1.3.1 Fallback to 3PCC Make Call

The uaCSTA MakeCall Request shall be extended to include the "Use3PCConError" private data extension to indicate fallback to 3PCC MakeCall (Section 4.6.1.17, "Make Call").

OpenStage device and OpenScape Voice V9 shall be enhanced to prompt the originating user before establishing a CSTA MakeCall based on the auto-answer configuration on the OpenStage device. If auto-answer is disabled, the phone will alert and the user must manually answer the call before the call is allowed to proceed to the b-leg.

The requirement to support 3PCC MakeCall even though the device supports uaCSTA MakeCall will result in increased media server usage to play ringback during call establishment. Solution team must plan for additional media server channels depending on the following factors, BHCA per subscriber, total number of subscribers. OpenScape Voice V9 Capacity planning tool should be used by service and planning team to determine the actual media server requirements for such a deployment with support from Systems Engineering.

1.3.2 One-way speaker call broadcast

A 1-way speaker broadcast call can be invoked via CSTA Make Call or Consultation Call using the broadcast PAC followed by the Group ID or using the broadcast PAC alone (in this case the group ID will be collected in-band).

CSTA flow direction is presented in the event connection information. The flow direction is shown in Originated, Established, Held and Retrieved events.

OSV privateData impacts defined in published privateData schema

- The privateData element "broadcast" is reported in Originated, Established, Held and Retrieved events for the initiator.
- The privateData element "Intercom" is reported in Established event for the receivers
- The private tag "broadcast" and the existing private tag "Intercom" shall be reported in the Snapshot Device response and Snapshot Call response for the initiator and the receivers accordingly.

Impacts on services permitted:

- Transfer Call, Single Step Transfer and Conference Call services are not allowed on 1-way speaker broadcast calls

- Deflect Call service is not allowed on one-way speaker broadcast call
- Consultation and Hold service are allowed on one-way speaker broadcast call
- It's possible to invoke a 3rd call leg after consultation from a 1-way speaker broadcast call. In this case the 1-way speaker broadcast call can be isolated (on hold) with no services until the third call leg is released.
- Alternate and Reconnect services are allowed during consultation from a 1-way speaker broadcast call
- If a device involved in a 1-way speaker broadcast call receives a second inbound call then Accept Call and Answer Call services are allowed on the inbound call

Refer to the following regarding the events reported and the CSTA services permitted on 1-way speaker broadcast calls:.

- The following service is required for 1-way speaker broadcast calls:
  - Section 4.6.1.17, "Make Call"
- The following services are allowed on 1-way speaker broadcast calls:
  - Section 4.6.1.9, "Consultation Call"
  - Section 4.6.1.14, "Hold Call"
  - Section 4.6.1.2, "Alternate Call"
  - Section 4.6.1.20, "Reconnect Call"
- The following services are not allowed on 1-way speaker broadcast calls:
  - Section 4.6.1.10, "Deflect Call"
  - Section 4.6.1.25, "Transfer Call"
  - Section 4.6.1.24, "Single Step Transfer Call"
  - Section 4.6.1.23, "Single Step Conference Call"
- A new privateData tag "**Broadcast**" is reported for the initiator in the following events:
  - Section 4.6.2.14, "Originated"
  - Section 4.6.2.8, "Established"
  - Section 4.6.2.10, "Held"
  - Section 4.6.2.16, "Retrieved"
- The privateData tag "**Intercom**" is reported in the following event for the receivers:

– Section 4.6.2.8, "Established"

1.3.3 Parallel - Simultaneous Alerting Model

This enhancement addresses limitations of the current OpenScape Voice Parallel hunting method which only allows a hunt group member to belong to 1 Parallel Hunt Group and not be a member of a Call Pickup Group. (Refer to the: CSTA Interface Specification Part II - CSTA Event Flow.)

1.3.4 CSTA change for hold/retrieve functionality

CSTA Interface option "Enable Consultation Call for Connections on Hold" can be set to extend the initial state requirement for Consultation Call defined in ECMA 269 to include connections on "Hold" (Refer to: Section 2.6.3.4, "Situational Considerations")

For applications that have the "Enable Consultation Call for Connections on Hold" option enabled, the servicesPermitted parameter includes "Consultation Call" and is sent in:

- All events where there is one call presently on hold.
- SnapShot request responses events where there is one call presently on hold.

When the interface option is disabled, then there is no change in the presentation of the Events currently presented for a connections in the "Hold" state.

1.3.5 Provide accumulated busy state for MLHGs / MLHG Pilot

OSV CSTA provides an "All Members Busy (AMB)" notification using a special Queued Event presented on monitored Hunt Group pilot.

For backward compatibility, the CSTA interface option "Enable All Members Busy Notification" has been created. The default for this option is "False" and must be enabled before OSV will send the special AMB Queued Event.

The AMB Queued Event is not associated with an actual inbound call and is reported independently from other inbound call Queued Events. The AMB notification Queued Event is only used to inform the application (e.g., OS UC) monitoring the Hunt Group pilot that all members are currently busy on a call and/or not available for calls (i.e., Hunt Make Busy is active).

If enabled, OSV shall send the AMB notification:

- When there is an active monitored on the Hunt Group pilot and all members are Busy on a Call or Hunt Make Busy is active.

- After a monitor is started on a Hunt Group where all members are currently Busy on a Call or Hunt Make Busy is active.
- After an AMB event has been sent OSV shall send a Clear Connection Event when at least one member becomes available and no other calls are currently in queue.

See QueuedEvent and Snapshot Device Response for xml example of AMB.

1.3.6 CSTA and OSMO/WebRTC Enhancements for Callback on Busy and No Reply

The following CallBack capabilities have been introduced:

- OSMO-enabled client invocation CSTA CallBack service when call is originated from client. This was previously blocked by Call Completion Service (CCS).
- Extend OSMO auto-pilot routing to support CallBack Recall by sending the setup first to the Client and then Desk Phone. No other OSMO devices will be selected for CallBack Recall.

  See CSTA Interface Specification PART II for event flow example.
- Introduce the CSTA Get CallBack service and response to enable client to display a list of pending callback targets. Client should provide option to cancel callbacks.

1.3.7 Centralized Call Log

OSV has introduced a new internal CSTA service process.

The first application to use the internal service is Centralized Call Log. This feature is designed to enable WebRTC enabled clients, like Ansible Connector Client to retrieve call log data directly from the switching function.

While the Call Log services and events use the CSTA transport layer they are defined in their own namespace outside of ECMA 323. New Call Log schema has been defined.

A Bulk Monitor Start service is introduce to enable the new process to implicitly start all monitors for either ONS-enabled or WebRTC/OSMO-enabled subscribers. The MonitorCrossRefID in CSTA events is presented as the subscriber's serviceID.

Call Log services and events are supported on CSTA over TCP and CSTA over SIP interfaces.

The Call Log Services:

- Call Log Snapshot - CF requests SF to all available call log data for specific

- Call Log Delete Entry - CF request SF to delete one or more call log entries

The Call Log Event:

- Call Log Event - SF send to CF to update call log information. This can be all call log entries (after Call Log Snapshot) or single entry in real time.

1.3.8 Do Not Disturb (DND) with Snooze Option

CSTA services and events listed below have been enhanced to enable the Do Not Disturb feature with a snooze duration capability to support the WebRTC client Snooze Notification feature from the OpenScape Voice platform:

- Set Do Not Disturb Service - Enables client to set DND with a snooze timer. When the snooze timer expires the DND feature is set to not active and DND event is sent.
- Get Do Not Disturb Response - Enables client to discover remaining snooze time, if any.
- Do Not Disturb Event - Informs all connected clients that the DND feature is active and if it has a snooze duration

1.3.9 CSTASM to extend the length of the appSessionID and block Reset of an active session

The applicationSessionID parameter length has been changed from 4 to 9 characters. In addition CSTA SM has added a check to block an application from "stealing" an applicationSessionID if it is in connected state.

1.3.10 Introduce a noReOffer value to the noReOffer privateData flag to prevent delay in call processing in certain scenario

Accept Call and Deflect Call service has been enhanced to prevent cases were call was unintendedly delayed. With this enhancement when a call fails:

- If the noReOffer string is <null> OSV will not long set the Offered Timer but immediately treat the call as Accepted.
- If the noReOffer string is "Retry" then OSV will reset the Offer Timer and allow the application to try to Accept or Deflect the call before OSV implicitly Accepts the call and proceeds.

Refer to CSTA I/F PART III - ONS (INTERNAL).

1.3.11 OpenScape_Voice V8 0 Branding

CSTA Get Switching Function Capabilities Response manufactureName change to "Unify".

1.3.12 Enable modification of CSTA Application Interface Options, with an appropriate warning, when a CSTA application status is connected

Administration: This CQ is associated with FRNs 6636, 6451, 6453, 7570 and 7626 V7R1 which required that application not be connected in order to change application's interface option. This restriction has been removed.

1.3.13 ODC WE, web client shown no green phone icon to accept the incoming video call when 3 incoming video calls left no answer

Currently if there is an existing alerting call is on a device and another call arrives and is also alerting then CSTA SM shall present "Answer Call" for subsequent calls alerting on the same device.

The behavior of the ODC WE Client is different than the OpenStage or OptiPoint phones that do not permit simultaneous multiple alerting connections. OpenStage does support a configuration that allow a "2nd Alerting" call, however the phone does not currently support 3PCC answer call on second call leg.

1.3.14 WebClient user cannot leave a callback against another party after not accepting a previous callback return call

Prior to this change a CSTA Clear Connection request invoked on a Callback - Recall connection was not cancelling the original callback request. This was not consistent with the behavior when the same call was rejected from the phone. This change provides CSTA parity with the phone behavior.

---Background--- CQ00227978 V7 - Was implemented to handle a CSTA Clear Connection request on any alerting connection as a reject (e.g., SIP 603 Declined).

1.3.15 CSTA-SM need to implement a random number for applicationSessionId and add a check/log entry for bad appSession records to the audit

Ensures unique session ID.

1.3.16 General Corrections and Updates

1.3.16.1 CSTA over HTTP/SOAP Interface

The limitation that only one connection is supported over CSTA over HTTP/SOAP has been removed from section 2.4.3.3.

No HTTPS support limitation has been added from section 2.4.3.3.

1.3.16.2 Application Interface Management

Added section (3.6) that describes CSTA application interface management.

1.4 References

1.4.1 External CSTA Documentation

Unless stated otherwise, supported OpenScape Voice CSTA services are compliant with the operational requirements described in the ECMA standards and technical reports listed here.

1. **European Computer Manufacturers Association (ECMA)**
   Computer Supported Telecommunications Applications (CSTA)
   http://www.ecma-international.org/activities/Communications/TG11/cstaIII.htm
2. **ECMA-269**
   Services for Computer Supported Telecommunications Applications (CSTA) Phase III, 7th edition (December 2006)
   http://www.ecma-international.org/publications/standards/Ecma-269.htm
3. **ECMA 354**
   Application Session Service
   First edition / June 2004.
   http://www.ecma-international.org/publications/standards/Ecma-354.htm
4. **ECMA TR/82**
   Scenarios for Computer Supported Telecommunication Applications (CSTA) Phase III, 2nd edition June 2009
   http://www.ecma-international.org/publications/techreports/E-TR-082.htm
5. **ECMA-323**
   XML Protocol for Computer Supported Telecommunications Applications (CSTA) Phase III
   4th edition (December 2006)
   http://www.ecma-international.org/publications/standards/Ecma-323.htm
6. **ECMA TR/72**
   Glossary of definitions and terminology for Computer Supported Telecommunications Applica¬tions (CSTA) Phase III
   3rd edition (June 2000)
   http://www.ecma-international.org/publications/techreports/E-TR-072.htm
7. **ECMA TR/87**
   Using CSTA for SIP Phone User Agents (uaCSTA)
   1st Edition / June 2004
   http://www.ecma-international.org/publications/techreports/E-TR-087.htm
8. **ECMA 348**
   Web Services Description Language (WSDL) for CSTA Phase III
   3rd. Edition
   http://www.ecma-international.org/publications/standards/Ecma-348.htm

1.4.2 Unify Documentation

- **OpenScape Voice Security Checklist**
  See eDoku for latest version.
- **OpenScape Voice SIP Interface Manuals**
  See eDoku for latest version.
- **OpenScape Voice Feature Interaction Matrix (FIM)**
  See eDoku for latest version.
- **Configuration Guide SIP Phones and Clients for CSTA**
  Contact GVS for latest version.

2 CSTA Service Overview

2.1 OpenScape Voice CSTA Operational Model

OpenScape Voice native CSTA interface is conformant ECMA-269 switching function CSTA service boundary.

OpenScape Voice functions in a limited fashion as a computing function. OpenScape Voice augments its native CSTA interface services with a non-conformant version of ECMA TR/87 SIP User Agent CSTA supported by the Unify OpenStage phone family.

2.2 ECMA CSTA Compliance Statement

Unless otherwise stated, OpenScape Voice CSTA services are compliant with the operational requirements described in the ECMA standards and technical reports listed in Section 1.4, “References”. This specification defines the details regarding specific support by OpenScape Voice.

2.2.1 ECMA-323 Namespace

OpenScape Voice supports ECMA-323 (Edition 4) Namespace and ECMA-269 (Edition 7).

2.2.2 ECMA-323 Specified Tags

The OpenScape Voice-internal CSTA XML encoder/decoder implements a portion of the ECMA-323 Edition 4 CSTA protocol as described in Chapter 4, “OpenScape Voice CSTA Service Description”.

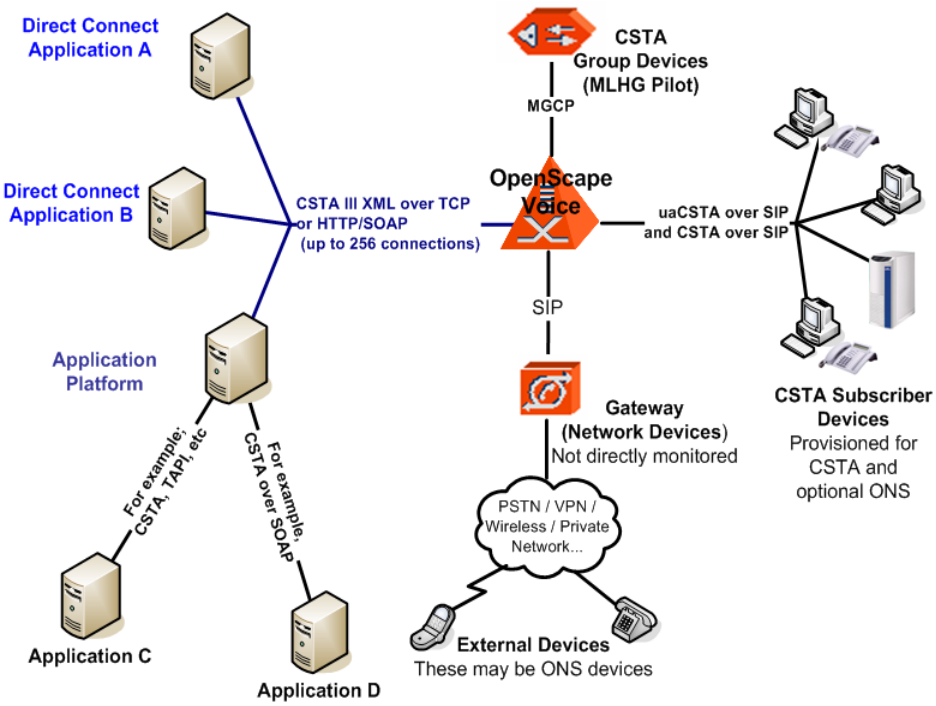
The OpenScape Voice XML schemas are defined in Appendix B, “OpenScape Voice privateData Schema”.

2.3 CSTA Application and OpenScape Voice Configurations

CSTA services are delivered to applications in one of two connectivity configurations. The figure below illustrates the two connectivity configurations supported by the OpenScape Voice CSTA interface:

- Application Direct Connect (Applications A and B).
- Application Platform, such as OpenScape UC Server (Applications C and D)

OpenScape Voice is able to support up to 256 simultaneous CSTA interfaces over TCP on 10/100mb Ethernet LAN protocol. Both application connectivity models are supported simultaneously. Applications are responsible for coordinating call control functions when the single device is monitored and controlled by multiple applications.



2.3.1 Direct Connect Configurations

In this connectivity configuration, applications establish the TCP connections and use the native OpenScape Voice CSTA interface directly. OpenScape Voice supports both direct and application platform connectivity models simultaneously. The direct connection interface is the application interface described in this specification.

2.3.2 CSTA Transport Considerations

2.3.2.1 CSTA over TCP

The OpenScape Voice CSTA TCP interface conforms to ECMA-323 – Appendix G – TCP without SOAP G.2 format.

The CSTA TCP IP address is provisioned in OpenScape Voice as described in Section 3.1, “Application Connectivity” and is known by the application by configuration or discovery using DNS SRV record.

Each CSTA application is responsible to establish CSTA connections on the provisioned TCP IP address.

The CSTA over TCP transport type requires a four-byte Invoke ID used by OpenScape Voice and application to correlate CSTA service request with response messages. The Invoke ID is provided in the TCP message body that precedes the XML instance message as illustrated in the following figure. The invoke ID is encoded as four ASCII numerical characters. A new Invoke ID is created in a CSTA service request message and is repeated in the corresponding service (positive or fault) response message. The value of “9999” is used for CSTA events.

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	0	Length		Invoke ID				XML message body

2.3.2.2 CSTA over SIP (for SIMPLE uaCSTA)

CSTA allows a Computing Function to invoke services on certain resources, typically a communications device as identified by a Directory Number (DN) or similar. A Switching Function executes these services and sends responses back to the Computing Function. A Switching Function can also send events reflecting changes of state of resources to the Computing Function at any time, based on a subscription by the Computing Function.

The uaCSTA interface specified in this document is designed to fit into a landscape where there are multiple Computing Functions, some residing in centralized applications and others residing on uaCSTA-enabled SIP clients such as phones or soft clients. Furthermore there is a Switching Function at the B2BUA and additional Switching Functions at uaCSTA-enabled SIP clients.

In a traditional CSTA environment, where uaCSTA is not involved, a Computing Function on a centralized application invokes services on and receives events from the Switching Function on the B2BUA (right hand side only of Figure 1 – Functional landscape with uaCSTA). The blue arrow represents CSTA communication for this purpose, where the single arrow-head indicates the direction of service requests and the double arrow-head indicates the direction of service responses and events.

In a uaCSTA environment in which a SIP client provides a Switching Function, certain CSTA services that are hard or impossible to execute at the Switching Function within the B2BUA are delegated by the B2BUA to the Switching Function in the SIP client. Examples are Make Call, Hold Call, Send Digits, Set Volume and Set Mute. Here the left hand side of the figure comes into play. The blue arrow represents CSTA communication for this purpose (uaCSTA), where the arrow-heads indicate service requests to the SIP client and services responses from the SIP client. This version of this document does not specify any events generated by the Switching Function in the SIP client.

Thus certain service requests from a Computing Function in a centralized application will be relayed by the B2BUA to the Switching Function in the SIP client, and responses will be sent in the reverse direction. Events will be sent by the Switching Function in the B2BUA to the Computing Function in the application, since the B2BUA will be able to see any resource state changes arising from actions at the Switching Function in the SIP client (e.g., it will see if a SIP call is established).

In a uaCSTA environment in which a SIP client provides a Computing Function, the SIP client is able to invoke certain CSTA services at the Switching Function in the B2BUA. In this version of this document, these are limited to getting and modifying B2BUA settings for call forwarding and DND for the user of the device concerned. When these resources change (e.g., by service request from a Computing Function on a SIP client, by service request from a central application, or by local action at the B2BUA), events will be sent to all interested Computing Functions, including that on the device concerned as well as any central application. The impact of this on the uaCSTA interface is shown by the red arrow on the left hand side of the figure. The red arrow represents CSTA communication for this purpose (uaCSTA), where the single arrow-head indicates the direction of service requests and the double arrow-head indicates the direction of service responses and events. Likewise the red arrow on the right hand side of the figure represents events to the central application arising from service requests by the SIP client.

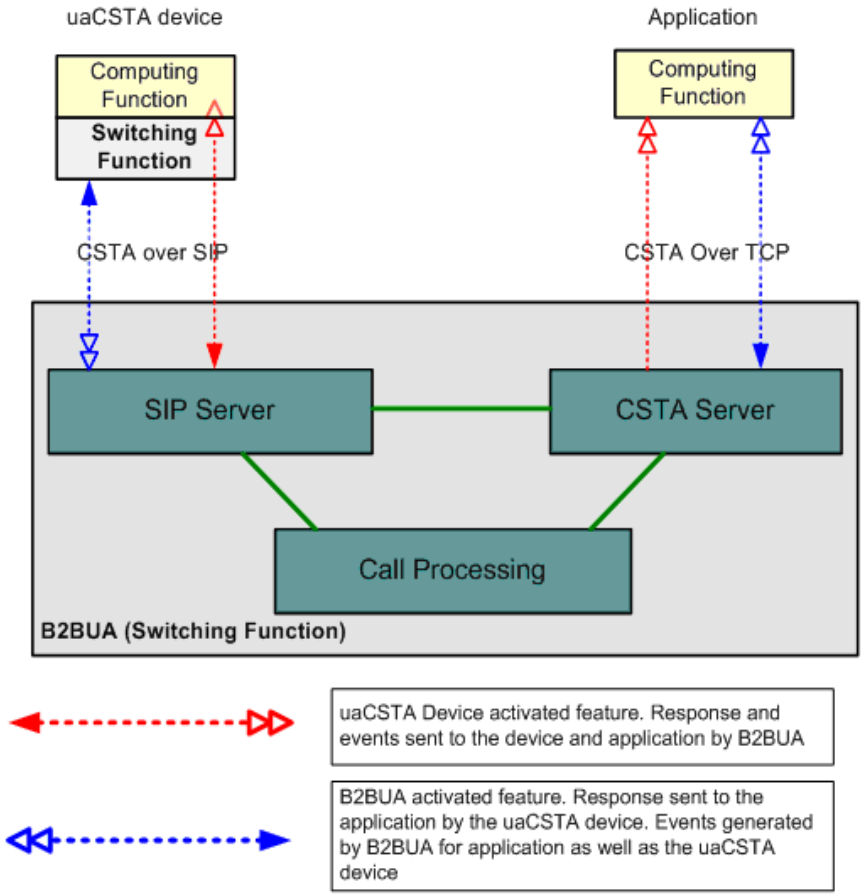


Figure 2 Simple usCSTA B2BUA functionality<

OpenScape Voice supports a limited User Agent CSTA (uaCSTA) interface with OpenStage phones. The interface is based upon the uaCSTA environment where OpenScape Voice is a back-to-back user agent fronting CSTA applications and using uaCSTA to communicate to Unify SIP phones on the back-end. The OpenScape Voice and OpenStage implementation is follows the uaCSTA landscape described in ECMA TR/87 and illustrated below.

CSTA does not use uaCSTA when multiple contacts are registered.

Refer to Section 2.4, “Supported CSTA Services and Events (by Device Type)” for a list of uaCSTA services supported by OpenScape Voice.

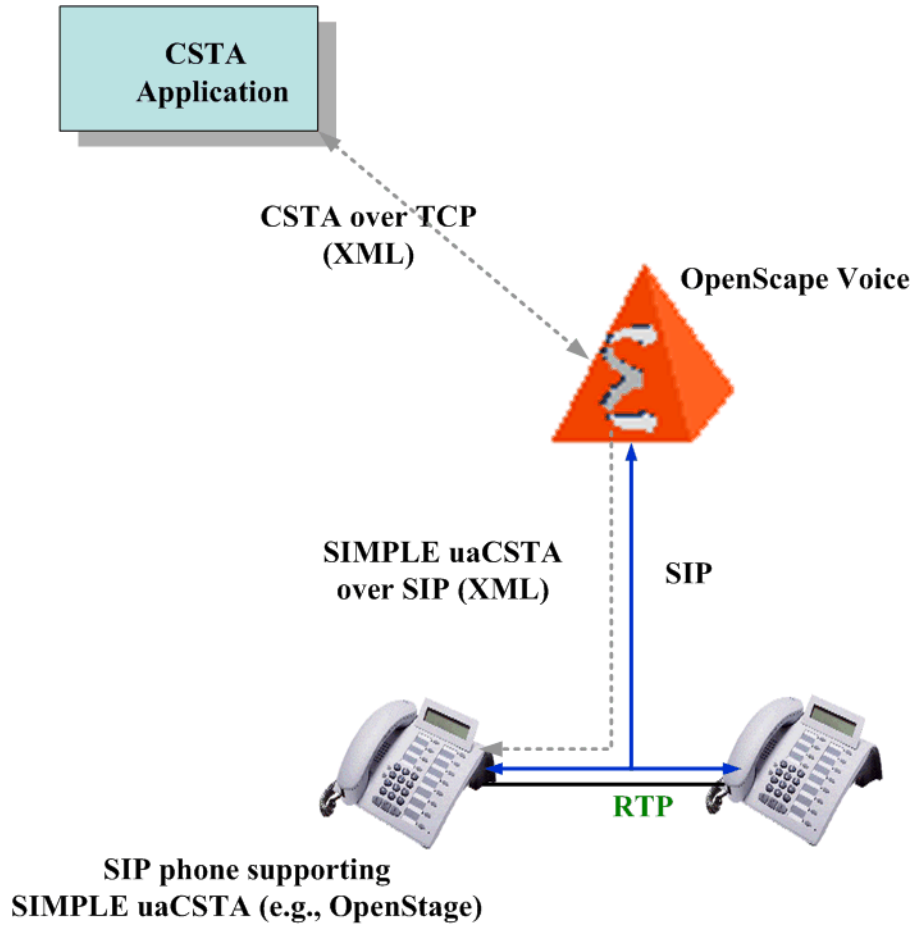


Figure 3 Simple uaCSTA connectivity

The uaCSTA is transported with SIP messages, but not within a specific SIP call dialog. This allows uaCSTA requests to be sent to the phone even if the phone is not currently involved in a call. CSTA over SIP transport is defined an internal OSCAR SIP User Agent specification.

OpenStage phones currently support CSTA over SIP for SIMPLE uaCSTA.

2.3.2.3 CSTA over HTTP/SOAP

CSTA over HTTP/SOAP based on ECMA 348 (edition 3) is supported.

Multiple CSTA applications can be served by a single OSV as long as each CSTA application uses a unique IP address for its CSTA link (for either HTTP/SOAP or TCP).

Note: ECMA 348 WSDL files reference ECMA 269 and 323 editions that are later than those supported by OpenScape Voice defined in section 2.2 of this specification.

Note: Currently OpenScape Voice does not support an HTTPS/SOAP interface.

2.3.3 CSTA over SIP

Refer to CSTA Interface Specification PART V (internal documentation).

2.3.4 QoS DSCP/TOS

QoS DSCP tagging of outbound CSTA packets is controlled by two RTP parameters defined in Table 1:

Parameter	Description	Type	Default Value
Srx/IpPacketPriority/DSCPTOS/CSTA/Type	Quality of Service (QoS) value for CSTA Signaling. This parameter determines if the value in Srx/IpPacketPriority/DSCPTOS/CSTA/Value is expressed as Differentiated Services Code Point (DSCP) or as Type of Service (TOS)	string	DSCP
Srx/IpPacketPriority/DSCPTOS/CSTA/Value	Quality of Service (QoS) value for CSTA Signaling. This parameter allows the setting of traffic class value (DSCP) or Type of Service value (TOS).	integer	26

Table 1 QoS DSCP/TOS

The acceptable integer values for Srx/IpPacketPriority/DSCPTOS/CSTA/Value RTP parameter are shown in Table 2.

Decimal DSCP Integer Values	DSCP Class	Precedence Level	Forwarding Type
10	1	Low	AF11
12	1	Medium	AF12
14	1	High	AF13
18	2	Low	AF21
20	2	Medium	AF22

Table 2 Integer Values for Srx/IpPacketPriority/DSCPTOS/CSTA/Value

Decimal DSCP Integer Values	DSCP Class	Precedence Level	Forwarding Type
22	2	High	AF23
26 (default)	3	Low	AF31
28	3	Medium	AF32
30	3	High	AF33
34	4	Low	AF41
36	4	Medium	AF42
38	4	High	AF43
40	5	-	Express Forwarding
46	Expedited	-	EF PHB (Expedited Forwarding)
0	Default PHB	Default	Best Effort

Table 2 Integer Values for Srx/IpPacketPriority/DSCPTOS/CSTA/Value

2.3.5 Application Link Security Considerations

2.3.5.1 CSTA Application White List

Access to OpenScape Voice CSTA services is secured by configuring valid and trusted server IP address table (that is, a “white list” of valid IP addresses). The CSTA Signaling Manager does not accept connections from IP addresses not configured in this list.

OpenScape Voice Element Management Command Line Interface (CLI) provides details on configuring denial of service options.

2.3.5.2 Transport Security using IPSec

IPSec is used to implement secure network layer connections between OpenScape Voice and CSTA Applications. Data transmitted over IPSec connections is subject to integrity and authentication checks and is encrypted to prevent eavesdropping. IPSec connections consist of two portions - a security policy that defines the level of security required on a subnet, host, port or protocol basis, and a security association that describes the algorithms and keys used to authenticate and encrypt data transmitted over the connection.

IPSec functionality is integrated into the network stacks of most operating systems. Utilities are provided for administration of IPSec related policies and associations. Key exchange is usually configured using the Internet Key Exchange (IKE) mechanism. An IKE daemon process, provided as part of IPSec system functionality, manages IKE exchanges and the corresponding setup of IPSec associations.

OpenScape Voice Element Management Command Line Interface (CLI) provides details on configuring IPSec Profile Management.

2.3.5.3 Transport Security via TLS (not supported)

TLS is not currently supported for CSTA TCP connections.

2.4 Supported CSTA Services and Events (by Device Type)

OpenScape Voice supports three (3) CSTA device type categories:

- Normal: for any non-Unify SIP device
- Type 1
- CSTA over SIP

Table 3 illustrates the recommended CSTA device type for SIP phones and clients.

CSTA Device Type	Unify Phones and Clients				Others
	OptiPoint 410 and 420	OpenStage 20, 40, 60, and 80	OpenScape Desktop Client with SIP Provider	OpenScape Desktop Client Personal or OptiClient 130	Any
Normal					X
Type 1[1]	X		X	X	
CSTA over SIP		X			

Table 3 Recommended CSTA Device Type for SIP Phones and Clients

1 Subscribers that do not have a physical registering device can still use CSTA and One Number Service (ONS Inbound and Outbound). In this case, it is recommended that Type 1 be provisioned.

2.5 Recommended CSTA Device Type Settings

Certain CSTA call and device control services are supported (X) by each CSTA device type. Table 4 illustrates the CSTA call and device control services supported by each CSTA device type.

Refer to Appendix C, “Supported CSTA Services and Events by Device Type” for a complete list of services and events supported by device type.

CSTA Device Type	Accept Call	Alternate Call	Answer Call	Clear Connection	Conference Call	Consultation Call	Deflect Call	Hold Call	Reconnect Call	Retrieve Call	Single Step Transfer Call	Transfer Call	Make Call	Generate Digits	Physical Device Caps e.g., Volume, Mute
Normal	X			X		X	X	X		X[1]	X	X	X[2]	X	
Type 1[3]	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
CSTA over SIP	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 4 CSTA Call and Device Control Services Supported by CSTA Device Types

1 If a call is placed on hold by the phone, then CSTA retrieve call will work only with Unify phones.
2 Auto-origination (hands-free) is not supported. This means that the calling device must go off-hook manually.
3 Subscribers that do not have a physical registering device can still use CSTA and One Number Service (ONS Inbound and Outbound). In this case, it is recommended that Type 1 be provisioned.

2.6 Common CSTA Parameter Considerations

This section defines the CSTA service, service response and event parameters supported by OpenScape Voice. As a general rule the conditional and optional parameters are only listed if they are supported by OpenScape Voice. The comment field provides specific usage information as it relates to OpenScape Voice. In most case this information is the same as ECMA-269.

The section references listed below apply to all OpenScape Voice CSTA services and events. These sections provide important details on specific parameter contents, formatting.

- **Supported Event Cause Codes** are defined in each CSTA event subsection description within Section 4, “OpenScape Voice CSTA Service Description”.
- **Supported Device ID Formats** are described in Section 2.6.1, “DeviceID Number and Name Presentation Conventions”.
- **Supported Negative Service Response Codes** are described in Section 2.6.2, “Negative Service Response”.
- **Supported Services Permitted** capabilities are described in Section 2.6.3, “Dynamic Feature Presentation (using servicesPermitted)”.

OpenScape Voice-specific support of CSTA Services is defined in the following sections. Services and events not listed are not supported.

2.6.1 DeviceID Number and Name Presentation Conventions

2.6.1.1 CSTA Device Identification Number

OpenScape Voice encodes the device identifiers (DeviceID) as described in the following sections.

OpenScape Voice supports the deviceID formats listed in the table below. DeviceID encoding the device identifiers (DeviceID) as described in the following subsections.

Device ID Format	Supported
Dialable Digits (DD)	Y
Switching Function Representation (SFR)	Y
Device Number	Y
URI	N

Dialable Digits (DD)

The DD format is used by an application in service requests to provide a sequence of dialing characters that is, used to reach a device. OpenScape Voice also uses the DD format in service responses and events and provides the following options:

OpenScape Voice supports the following options for the DD format:

- Supports characters “0” – “9”; characters “*”, “#” and “+”. The comma is supported, but only for the Generate Digits service.
- Permits visual formatting characters such as space, “(“, “)” and “-“from the application. However, visualization formatting characters are ignored and removed by OpenScape Voice signaling manager and not repeated in any service response or events.
- International number format (with leading plus “+”) is supported by OpenScape Voice as a dialable number format. The presentation of international number format without visual formatting characters is also referred to as Global number format (GNF).
- OpenScape Voice may send DD format in CSTA responses and events when no other deviceID information is available, such as, Name, Display Number or OND.

Note: One Number Service Device (OND) is internal to Unify and is used over the CSTA interface between OpenScape Voice and OpenScape UC Application (refer to Appendix F, “One Number Service (ONS)” for further details).

Examples of DD Formatted DeviceIDs:

- 22343 – a sequence of characters that can be dialed to reach a station. This maybe a fully qualified private network number.
- +1 (123) 555-1212 – a sequence of characters representing an international number (with visual separators)
- 914084922343 – a sequence of characters that includes an access (exit) code to reach an external device
- +15613221234 - a sequence of characters representing a global or international number format.

Switching Function Representation (SFR)

SFR format is a sequence of characters that is, used to reference a device including a directory number, name, and other value added specified addressing attributes supported by OpenScape Voice.

This SFR format may be used in service requests, service responses, and events.

Switching Function Representation (SFR) may be provided when name and number information is provided. OpenScape Voice conforms to SFR format as defined in section 10.1.2 of ECMA-269.

The SFR deviceID is identified by a leading “N” and has the following format:

N<DN/PE>NM;tag=value;tag=value

Table 5 identifies the supported and not supported SFR fields and tags:

Field Name	Supported	Options
Directory Number (DN)	Y	Digits “0” through “9”, “*”. “#”, “+” and readability characters: “-“, “(“, “)”. All other characters shall be ignored. Refer to the Examples of DD Formatted DeviceIDs:. A leading “+” indicates international or global number format Fully qualified private numbers are also supported with proper provisioning. Refer to Appendix D. Examples of DN formatted deviceIDs: • 22343 - a sequence of characters representing a fully qualified private network number (PNP) or extension that can be dialed to reach an internal or on-net subscriber device. • +1 (123) 555-1212 - a sequence of characters representing an international number (with visual separators) • 914084922343 - a sequence of characters that includes an access (exit) code to reach an external device • +15613221234 - a sequence of characters representing a global or international number format. • ABC1 – used to for Park to Server. The display name “Park Slot” is included along with the displayNumber (see below) that contains the parking space
Physical Element (PE) Extension String (/)	N	Service request with this field will be rejected.

Table 5 SFR Fields and Tags

Field Name	Supported	Options
Name String (NM)	Y	• The NM field begins with a sequence of characters that represents the name of the person associated with the Directory Number. OpenScape Voice only supports calling party name presentation. This field is provided by OpenScape Voice when available. • OpenScape Voice supports multibyte characters primarily for the Chinese market. See Section 2.6.1.5, “Multibyte Support for Name (NM) Field in UTF-8 Encoding”. • ; (semicolon) is the separator character that precedes a tag name. • tag is the specified name of the Name String tag (Tags supported by OpenScape Voice are defined in Table 6). • =value is the assigned value of the tag. Depending upon the tag, this portion may be omitted. Tag and value pairs are repeated in the NM as necessary. OpenScape Voice assumes that if an application does not use a specific tag that it simply ignores it without consequence.

Table 5 SFR Fields and Tags

The following Name String tags are supported by OpenScape Voice. OpenScape Voice ignores any unsupported tag.

Field Name	Supported	Options
displayNumber	Y	Indicates an optimal dialable form of the Directory Number (e.g. extension number, public number with route access code, etc.). This notation is typically a dialable and displayable number used in call information displays and call logs. OpenScape Voice does not provide the displayNumber parameter for deviceIDs portion of a connectionID. The Phone and CSTA number display information should be synchronized. The following rules apply: • If display information is provided to the phone, the CSTA display information must match the display of the phone. • If display information has not been provided to the phone, CSTA attempts to provide display information using number modification API when privacy restrictions are not applicable to the party being displayed. One example of this situation is for Inter-BG call from A to B, where party A does not subscribe to number presentation and B does not have privacy set for Inter-BG calls. In this case, A's phone shows dialed digits, not a modified number. The CSTA Originated event also reflects the dialed digits. However, in the CSTA Delivered event and beyond, the SFR format displayNumber for the called party is the modified number analysis. Refer to OpenScape Voice Assistant – Display Number section for details on provisioning of the number definition table. --- **Note:** If the device number (DN) field is ABC1, then the display number is set to the current parking space number. ---

Table 6 Supported SFR Tags *(Seite 1 von 2)*

Field Name	Supported	Options
ond	Y	Indicates the sequence of dialable characters used to reach a One Number Service Device (OND) associated with the user's ONS-published number (represented in the Directory Number field). For ONS DeviceIDs, the OND represents the actual device used by the ONS user (and shall be a private number), whereby the Directory Number field represents the user's published number, which is the number exposed to its call partners. The OND is supported in the same formats as the Directory Number field. **Note:** OND is internal to Unify and is used over the CSTA interface between OpenScape Voice and OpenScape UC Application (refer to Appendix F, "One Number Service (ONS)" for further details).
exclude	Y	The exclude tag was implemented for a specific Microsoft Office Communicator (MOC) implementation for multiple contacts. If the tag value is moc (exclude=moc) in the Make Call or Answer call service then OpenScape Voice shall not be considered the registered MOC as an associated device for the Make Call or Answer Call request.
keyID	Y	The keyID tag appears in connectionIDs where shared keyset lines are in use. The keyID tag contains the deviceID of a device currently using an appearance of the primary DN. The keyID tag ensures that connectionIDs are unique during conference (bridge) and transfers.
uid	Y	The uid tag appears in the connectionID when the calling or called device is external to OSV. The uid tag ensures that connectionIDs are unique during conference (bridge) and transfers.
noa	Y	The noa tag is presented if the presented deviceID has failed to produce an FQN. The value of noa in this case is "nk".

Table 6 Supported SFR Tags *(Seite 2 von 2)*

Examples of SFR formatted deviceIDs:

```
N<+1(408)492-2343>John
Smith;ond=+14085551212;displayNumber=22343
```

```
N<+1(408)492-2343>;ond=+14085551212;displayNumber=22343
```

```
N<+1(408)492-2343>;prNum
```

```
N<+1(408)492-2343>;appCallbackID=+1(408)492-
2000;appCallbackName=Accounting Dept
```

2.6.1.2 Global Number Format

- Global Number Format (GNF) supports both the DD and SFR (DN) device ID formats.
- GNF is identified by a leading plus (“+”) and not other formatting characters.
- GNF impacts all supported services, service responses and events.
- Applications service requests using GNF deviceID format are identified internally within OpenScape Voice as international numbers. If GNF deviceID format is not used by the applications the provided number is treated as an “unknown” type within OpenScape Voice. Unknown numbers are assumed to be dialable for the purposes of routing or the service ID of an internal subscriber or subscriber extension.
- OpenScape Voice supports GNF format in non-call-related services.

2.6.1.3 Device Number (DeN) Format

The provisioning Endpoint page in CLI and OpenScape Voice V9 Assistant present the CSTA NID deviceID.

The NID shall be presented in the Device Number (DeN) format. The DeN format is a non-dialable 4-byte integer value. OpenScape Voice V9 uses the most significant byte to identify the device type and the least significant 3-bytes for the device number. The device number shall be based on the least significant 3-bytes of the existing internal endpoint identifier (EPID) assigned in the endpoint database record. The internal EPID is updated by one each time an endpoint device is added to the system.

Note: EPID values are not re-used when an endpoint is deleted.

MSB 1 byte	LSB 3 bytes		
Device Type Indicator	DeN	DeN	DeN

Table 7 Device Number (DeN) Format

The Network Interface Device (NID) type and Device Number ranges supported by OpenScape Voice V9 are shown in Table 8.

Device Type	Device Type Mask (MSB)	Internal EPID Range	Device Number Display Range
Network Interface Device (i.e. Trunk gateway)	Hex: 08000000 Dec: 134,217,728	Hex: 000001 - FFFFFF Dec: 1 - 16,777,215	Hex: 08000000 - 08FFFFFF Dec: 134,217,729 – 135,894,943

Table 8 Network Interface Device and supported Device Number ranges

Device	Device Identifier Format	Device Number Range	Device Number Device Type in MSB
Extension	Diallable Digits / SFR	NA	NA
ACD agent	Diallable Digits / SFR	NA	NA
Attendant Console	Diallable Digits / SFR	NA	NA
ACD Group Number	Device Number	1-255	1
ACD RCG Number	Device Number	1-1020	2
Announcement	Device Number	1-64	3
General Attendant	Device Number	0-999999	4
Hunt Group	Device Number	0-999999	5
Music	Device Number	1-32	6
Parking Slot	Device Number	0-9	7
Trunks	Device Number	0-9999	8
PhoneMail trunk	Device Number	0-9999	8
Special device	Device Number	0	10

For deviceID parameters that only present the NID, such as, associatedCallingDevice, associatedCalledDevice and device (as used in the Call Information Event) the attribute typeOfNumber = “deviceNumber” must be included in order for the application to correctly identify that the number is not a dialingNumber (default for OpenScape Voice V9).

Example XML with deviceNumber:

```
<associatedCalledDevice typeOfNumber=”deviceNumber”>
      <deviceIdentifier>N&lt;134217729&gt;</deviceIdentifier>
</associatedCalledDevice>
```

2.6.1.4 Number and Name Display Including Privacy

1. OpenScape Voice currently does not support deviceID name (prNam) and number (prNum) privacy tags defined in the SFR format section. Refer also to Switching Function Representation (SFR) in Section 2.6.1.1.
2. OpenScape Voice supports several privacy subscriber features that will block name and number information. Privacy applies to subscriber DNs and Group Devices (MLHG Pilot) DNs.
   – Privacy features may be applied to calling and called number presentation in either the DD or SFR formats.
   – Privacy features may be applied to calling and called number and name presentation independently when using the SFR format. For example, number presentation may be blocked and only the name presented.
3. The following privacy presentation are applied to the device identifiers for the following scenarios:

- Subscriber Number Privacy = “Restricted”. Applies to DD or SFR (DN field and displayNumber tag).
- Subscriber Name Privacy = blank. Applies to SFR name portion of NM field. All other tags follow normal presentation rules. Example of deviceID with blank name: N<+498972249995>; displayNumber=49995.
- External Number Privacy (Out-of-area or Private) = “Restricted” “. Applies to DD or SFR (DN field and displayNumber tag). Out-of-Area applies for inter-BG and gateway calls. Privacy applies when number blocking services are applied.
- External Name Privacy (Out-of-Area or Private) = “Not Known”. Applies to name portion of NM field only. All other tags follow normal presentation rules. Out-of-Area applies for inter-BG and gateway calls. Privacy applies when number blocking services are applied.

4. Group Device (MLHG) number display considerations:
   - For calls to an MLHG pilot number the CSTA events reported to the caller will provide the number display info of the MLHG pilot for the agent device. If the MLHG pilot has subscriber number privacy set, the display info associated with the agent device will also be “restricted”. If the MLHG pilot does not have privacy set, the display info associated with the agent device will be the display info that is associated with the pilot.
   - For calls to an MLHG pilot number, the CSTA events reported to parties other than the caller (i.e. the pilot or the agent) will provide the display info of the agent for the agent device, and this display info will reflect the privacy settings for the agent.
5. CSTA supports sending party’s identities in the CSTA events. The party’s GNF or fully qualified private number will be used as device identifier. Refer to Section 2.6.1.2, “Global Number Format” and Section 3.2.8, “Global Number Format (GNF) Replacement - RTP Parameter”, for additional details.
6. CSTA also reports, via the displayNumber name field tag, the formatted display information for the parties reported in the events, except for the reporting device itself. The display information contains the name and the ‘dialable’ display number as created via the OpenScape Voice Display Number Modification tables.
7. The display information is optional for CSTA Conferenced Events and subsequent events for the Conferenced connections (i.e. Failed, Held, and Retrieved).
8. OpenScape Voice call processing has a restriction where the phone displays are not updated for External calls. However, for CSTA, the OpenScape Voice does report the normal CSTA events with updated display information for the life of the call.

9. OpenScape Voice has implemented an exception to the “Restricted” deviceID for conferencingDevice. This deviceID is always presented. It is the applications responsibility to restrict presentation of this parameter.

CSTA Number and Name Privacy Presentation Rules

Display	Internal / External	CSTA deviceID privacy	Example deviceID format
Number	Internal *	The device identifier status is presented as “restricted” and the displayNumber tag is omitted.	N<restricted>Jim Smith
	External	Same as internal.	N<restricted>Boca
Name	Internal	The name portion of the deviceID is empty (blank).	N<+498972249995>; displayNumber=49995
	External (BG)	The name portion of the deviceID is as provided by gateway or the provisioned BG name is provided.	N<+15615551212>Boca; displayNumber=81561555 1212
	External (E.164)	The name portion of the deviceID is empty (blank). *OPEN: Verify with development if “Not Know” is used as described in the CSTA Interface Specification.*	N<+15615551212>; displayNumber=81561555 1212
***** Case 1; For calls to an MLHG pilot number, the CSTA events reported to the caller provide the number display info of the MLHG pilot for the agent device. If the MLHG pilot has subscriber number privacy set, the display info associated with the agent device is also “restricted”. If the MLHG pilot does not have privacy set, the display info associated with the agent device is the display info that is associated with the pilot. Case 2: For calls to an MLHG pilot number, the CSTA events reported to parties other than the caller that is the pilot or the agent provide the display info of the agent for the agent device, and this display info reflects the privacy settings for the agent.			

Suggested rules for CSTA application number display in UI:

- Always display the information in the displayNumber tag, if present.
- If displayNumber tag is not present and deviceID is not “restricted” then the deviceID number may be presented.
- If the deviceID number is restricted then application should provide language specific or language neutral information that indicates that the number is “Private”.
- If an OND tag is presented it should NEVER be presented as a display number to the user.

Note: One Number Service Device (OND) is internal to Unifyand is used over the CSTA interface between OpenScape Voice and OpenScape UC Application (refer to Appendix F, “One Number Service (ONS)” for further details).

Suggested rules for CSTA application name display:

- If the application does not support a directory name lookup by number function then the application should always display the information provided in the name field of the deviceID.
- If the application does support a name lookup function by number and the deviceID is not “restricted” then the name may be display as it appears in the directory. However, if no entry is found in the directory then the name field of the deviceID should be displayed.
- If the application receives a CallInformationEvent on a device monitor after an Offered, Delivered, Queued or Established event, the callingDevice parameter may include an updated calling device name.

2.6.1.5 Multibyte Support for Name (NM) Field in UTF-8 Encoding

OpenScape Voice supports multibyte characters primarily for the Chinese market. With this multibyte implementation, OpenScape Voice potentially can support other foreign languages in the future. The multibyte support is limited to DisplayName and BusinessGroupLineName sent to SIP endpoints and CSTA-enabled applications.

Only selected subscribers’ data visible to the phone and CSTA application users is in UNICODE. Subscribers’ Chinese display-name needed by SIP endpoints or CSTA applications is provided. All system administrative data such as Business Group and Class Of Service etc. and all provisioning needed to make the system work remain as it is today.

When subscriber device names are provisioned for UNICODE format CSTA sends out calling party display names in the Delivered and Established events in UTF-8 encoding.

Since CSTA events are formatted in UTF-8 encoded XML, any character with a value exceeding 127 (in addition to some characters used in the XML syntax) is currently represented in an escaped sequence. Since a UTF-8 encoded character string could conflict with this escaping mechanism, the entire name string is now expected in UTF-8 encoded format and the escaping mechanism is no longer be applied, except to XML syntax specific characters such as ‘<’, ‘>’, ‘/’, and so on.

2.6.2 Negative Service Response

The following service response error codes are supported by OpenScape Voice. The “x” mark in an error code column indicates that the error code is possible.

	Operation Errors														All Other Errors								
Error Values -> **CSTA Service**	**generic**	**invalidDevicID**	**invalidCallID**	**privellegeViolationSpecificDevice**	**privellegeViolationSpecificDevice**	**InvalidMonitorObject**	**invalidMonitorCrossRefId**	**requestIncompatibleWithMedia**	**requestIncompatibleWithDevice**	**requestIncompatibleWithMedia ActiveConnection**	**serviceNotSupported**	**InvalidDestination**	**InvalidForwardingDestination**	**invalidForwardingType**	**generic "category"**	**deviceOutOfService**	**invalidConnectionState**	**resourceBusy**	**requestedProtocolVersionNotSupported**	**noSessionIDAvailable**	**serverResourcesBusy**	**maxNumberSessions**	**invalidSessionID**
CAPABILITIES EXCHANGE SERVICES																							
get CSTA features	x														x								
get logical device Information	x	x		x											x								
get switching function capabilities	x														x								
SYSTEM SERVICES																							
system register	x														x								
request system status	x														x								
system status	x														x								
MONITORING SERVICES																							
monitor start	x	x													x	x							
monitor stop	x														x								
SNAPSHOT SERVICES																							
snapshot call	x	x	x	x																			
snapshot device	x	x										x			x	x							
snapshot devicedata	x	x													x								
APPLICATION SESSION SERVICES																							

Table 9 Supported Negative Service Responses

	Operation Errors														All Other Errors								
Error Values -> **CSTA Service**	**generic**	**invalidDevicID**	**invalidCallID**	**privellegeViolationSpecificDevice**	**privellegeViolationSpecificDevice**	**InvalidMonitorObject**	**invalidMonitorCrossRefId**	**requestIncompatibleWithMedia**	**requestIncompatibleWithDevice**	**requestIncompatibleWithMedia ActiveConnection**	**serviceNotSupported**	**InvalidDestination**	**InvalidForwardingDestination**	**invalidForwardingType**	**generic "category"**	**deviceOutOfService**	**invalidConnectionState**	**resourceBusy**	**requestedProtocolVersionNotSupported**	**noSessionIDAvailable**	**serverResourcesBusy**	**maxNumberSessions**	**invalidSessionID**
abort application session																							x
reset application session timer																							x
start application session																			x	x	x	x	x
stop application session																							x
CALL CONTROL																							
accept call																							
alternate call	x		x												x								
answer call	x		x												x								
call back call-related	x	x		x				x	x														
clear connection	x	x	x												x								
conference call	x		x												x								
consultation call	x		x												x								
deflect call (target is alerting or offered party)	x		x							x	x	x											
directed pickup call	x	x	x	x											x	x	x						
group pickup call	x	x		x											x	x	x						
hold call	x		x												x								
make call	x	x										x			x	x							

Table 9 Supported Negative Service Responses

	Operation Errors														All Other Errors									
Error Values -> **CSTA Service**	**generic**	**invalidDevicID**	**invalidCallID**	**privellegeViolationSpecificDevice**	**privellegeViolationSpecificDevice**	**InvalidMonitorObject**	**invalidMonitorCrossRefId**	**requestIncompatibleWithMedia**	**requestIncompatibleWithDevice**	**requestIncompatibleWithMedia ActiveConnection**	**serviceNotSupported**	**InvalidDestination**	**InvalidForwardingDestination**	**invalidForwardingType**	**generic "category"**	**deviceOutOfService**	**invalidConnectionState**	**resourceBusy**	**requestedProtocolVersionNotSupported**	**noSessionIDAvailable**	**serverResourcesBusy**	**maxNumberSessions**	**invalidSessionID**
reconnect call	x		x												x									
retrieve call (from hold)	x		x												x									
single step transfer	x	x	x									x			x									
transfer call	x		x												x									
CALL ASSOCIATED FEATURES																								
change connection information	x		x				x								x									
generate digits	x		x												x									
PHYSICAL DEVICE FEATURES																								
get message waiting indicator	x	x													x	x								
get microphone mute	x	x		x			x								x			x						
get speaker volume	x	x		x			x								x			x						
set microphone mute	x	x		x			x								x			x						
set speaker volume	x	x		x			x								x			x						
LOGICAL DEVICE FEATURES																								
call back non-call-related	x	x		x				x	x						x									
get agent state	x	x													x									
get do not disturb	x	x		x				x							x									

Table 9 Supported Negative Service Responses

	Operation Errors														All Other Errors									
Error Values -> **CSTA Service**	**generic**	**invalidDevicID**	**invalidCallID**	**privellegeViolationSpecificDevice**	**privellegeViolationSpecificDevice**	**InvalidMonitorObject**	**invalidMonitorCrossRefId**	**requestIncompatibleWithMedia**	**requestIncompatibleWithDevice**	**requestIncompatibleWithMedia ActiveConnection**	**serviceNotSupported**	**InvalidDestination**	**InvalidForwardingDestination**	**invalidForwardingType**	**generic "category"**	**deviceOutOfService**	**invalidConnectionState**	**resourceBusy**	**requestedProtocolVersionNotSupported**	**noSessionIDAvailable**	**serverResourcesBusy**	**maxNumberSessions**	**invalidSessionID**
get forwarding	x	x		x										x	x			x						
set agent state	x	x													x									
set do not disturb	x	x		x											x			x						
set forwarding	x	x		x									x	x	x			x						

Table 9 Supported Negative Service Responses

Note: The negative acknowledgement error codes sent by the computing domain are application specific and is ignored by OpenScape Voice. An entry in the error log is generated.

2.6.3 Dynamic Feature Presentation (using servicesPermitted)

2.6.3.1 General

OpenScape Voice makes every effort to provide accurate dynamic feature presentation data however 100% accuracy of the servicesPermitted parameter is not guaranteed.

Support with servicesPermitted is restricted to indicating CSTA Call Control and CSTA Snapshot services. Support on events is limited to CSTA Call Control and CSTA Call Associated events.

Figure 4 shows how servicesPermitted parameter is determined by OpenScape Voice. The inverse pyramid represents all possible CSTA services defined by ECMA 269. The bands within the pyramid represent from top to bottom items A-D are considered mostly static services permitted and E-F mostly dynamic services permitted.

- CSTA services currently supported by OpenScape Voice are listed in the CSTA Interface Specification. These are all possible statically available services.
- With the possible exception of OpenScape Voice customers, subscriber features may be purchased A la Carte by the provider or provisioned as part of a profile. Some feature provisioning statically available CSTA services. For example, a Business Connection must purchase CSTA, ONS, Forwarding, MLHGs, and so on. The CSTA interface specification identifies the OpenScape Voice features that affect CSTA services. Refer also to Section 3.2.1, “BGL Services Affecting CSTA Services”.
- OpenScape Voice providers may resell services to their customers which may affect available CSTA services.
- Subscriber, including OpenScape Voice, may enable or disable certain features which may affect available CSTA services.
- During the course of a call only certain CSTA services are permitted depending on the event, connection state and possibly the cause.
- Even in a specific call state, certain restrictions come about because of other conditions, for example, number of calls at a device, and so on. These can be applied as special rules after obtaining the basic rule based on event, connection state and cause.

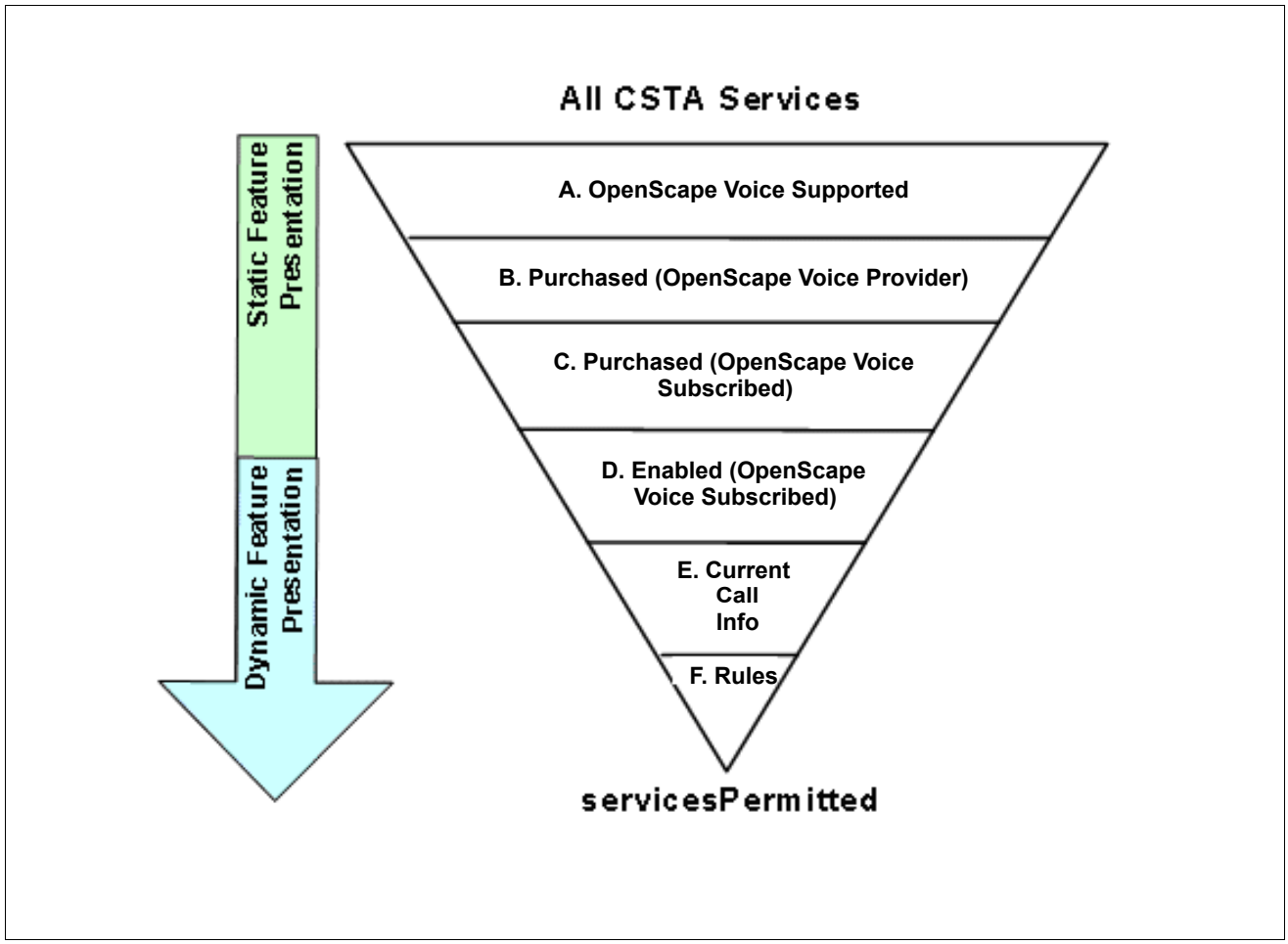


Figure 4 servicesPermitted Inverse Pyramid

Layers A through F are combined to create the set of services that can be executed upon a call in a given call state. To be available to a user, a feature must be permissible in each of the layers.

Within layers B, C and D, purchased and enabled features have an effect on the static feature presentation, and thus later on the dynamic aspect of the services permitted.

Within layer E is affected by device type where service is invoked (Table 10).

2.6.3.2 Device Type Considerations

CSTA Service→ **Standard CSTA by Device Type**	**Accept Call**	**Alternate Call**	**Answer Call**	**Call Back Call-Related**	**Clear Connection**	**Conference Call**	**Consultation Call**	**Deflect Call**	**Hold Call**	**Reconnect Call**	**Retrieve Call**	**SS Transfer Call**	**Transfer Call**	**Change Conn. info**	**Generate Digits**	**Make Call**
Normal	✓			✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓
SimenesType1 Phone / Client	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CSTA over SIP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MLHG Pilot (Group Device)					✓			✓								

Table 10 servicesPermitted by Device Type

2.6.3.3 Call State Considerations

Deeper within **layer E**, the state of the connection affects the servicesPermitted in a dynamic manner. Table 11 servicesPermitted by Connection State provides a list of currently supported events and the servicesPermitted presentation. The table bases servicesPermitted presentation on the local connection state of the device receiving the event. In considering the services permitted, the event cause may also need to be considered when determining the feature availability. This table only represents the basic results as further rules is used to modify what the application sees.

CSTA Service → / **Connection State**	**Accept Call**	**Alternate Call**	**Answer Call**	**Call Back Call-Related**	**Clear Connection**	**Conference Call**	**Consultation Call**	**Deflect Call**	**Hold Call**	**Reconnect Call**	**Retrieve Call**	**SS Transfer Call**	**Transfer Call**	**Change Conn. info**	**Generate Digits**	**Make Call**
Alert – offering	**Yes**	N[2]	N[2]	**Yes**	**Yes**	N[1]	N[1]	**Yes**	N[1]	N[2]	N[1]	N[2]	N[1]	N[1]	N[2]	N[2]
Alert – ringing	N[1]	**Yes**	**Yes**	**Yes**	**Yes**	N[1]	N[1]	**Yes**	N[1]	**Yes**	N[1]	N[2]	N[1]	N[1]	N[2]	N[2]
Alert – entering distribution	N[1]	N[2]	N[2]	N[2]	N[2]	N[1]	N[1]	N[2]	N[1]	N[2]	N[1]	N[2]	N[1]	N[1]	N[2]	N[2]
Connected	N[1]	**Yes**	N[1]	N[1]	**Yes**	**Yes**	**Yes**	N[2]	**Yes**	**Yes**	N[1]	**Yes**	**Yes**	**Yes**	**Yes**	N[2]
Failed	N[1]	N[1]	N[1]	**Yes**	**Yes**	N[1]	N[1]	N[2]	N[1]	N[2]	N[1]	N[2]	N[1]	N[1]	N[2]	N[2]
Hold	N[1]	**Yes**	N[1]	N[1]	**Yes**	**Yes**	N[1]	N[2]	N[1]	**Yes**	**Yes**	N[2]	**Yes**	**Yes**	N[2]	N[2]
Initiated	N[1]	N[1]	N[2]	N[1]	**Yes**	N[1]	N[1]	N[1]	N[1]	**Yes**	N[1]	N[1]	N[1]	N[1]	N[2]	N[2]
Null	N[1]	N[1]	N[1]	N[1]	N[1]	N[1]	N[1]	N[1]	N[1]	N[1]	N[1]	N[1]	N[1]	N[1]	N[1]	**Yes**
Queued	N[1]	N[2]	N[2]	N[2]	**Yes**	N[1]	N[1]	**Yes**	N[1]	N[2]	N[1]	N[2]	N[1]	N[1]	N[2]	N[2]

Yes = Permitted by CSTA and OpenScape Voice
N[1] = Not permitted by CSTA
N[2] = Not permitted by OpenScape Voice

Table 11 servicesPermitted by Connection State

2.6.3.4 Situational Considerations

The final layer of the inverse pyramid, **layer F**, represents as set of exception rules which are used to clarify the final services permitted value. Some of the key conditions are as follows:

- CSTA permits a **Make Call** when the device can support a new Null or Initiated call. It does just depend upon the current connection state of the device per se, but the ability of the device to support multiple calls in that state.
- **Reconnect** requires that one connection be in Hold state and there is more than one call on the device else the service is not permitted.
- **Alternate** requires that one connection be in Connected state else the service is not permitted.
- **Transfer** requires both a connection in Hold state and a call in Connected state else the service is not permitted. The device cannot be part of a conference else the service is not permitted.
- **Conference** requires both a connection in Hold state and Connected state else the service is not permitted.
- **Change Connection Information** requires an established state for one party, which means at least one party is in connected state and the other is in Connected or Held state.
- **Single Step Transfer** requires **only one call** be present for the device in Connected state else the service is not permitted. The service is also not permitted if the device is part of a conference (TWC or Large).
- **Generate Digits** requires a call in a connected state that is, not part of a conference.
- The only service allowed in connected state before an established event is received is **Clear Connection** or **Reconnect Call**. From the A side the Originated and Delivered are both sent in the connected state.
- If the current call leg is part of a Large Conference (SIP), the only allowed services are Consultation, Hold, Retrieve, Reconnect, Alternate, ClearConnection and Conference. This call never digresses back to a basic call.
- **Consultation Call** will be presented on the active call. If the physical device supports uaCSTA, then a consultation call will be presented when there are two call legs active. If there are multiple calls at an AoR with multiple contacts (see Section 3.2.7, “Multiple Contacts (SIP Forking)”), then consultation is possible on only one callID.

  - Applications with the "Enable Consultation Call for Connections on Hold" attribute enabled can perform Consultation Call requests for a connection in the “Hold” state.The servicesPermitted parameter shall include "Consultation Call" in the Held Event or Call Information Event (related to a held connection).

- **Retrieve** does not work for non-uaCSTA devices unless the device was put on hold using the application.
- **Answer Call** is not allowed if multiple contacts are registered on the called device.
- **Call Back Call-Related** service is not presented in A-side Delivered Event if the cause code is:

  a) ENTERING_DISTRIBUTION

  b) RECALL

  c) RECALLBUSY

  d) RECALLFORWARDED

  e) RECALLNOANSWER

  f) CALLBACK or

  g) SINGLESTEPTRANSFER

2.6.3.5 Static Condition Considerations

- **Answer Call** does not work unless the subscriber is provisioned as Type1 or CSTA over SIP.
- **Alternate Call** will be presented but will not work if the autoReconnect feature is not configured on the Unify phone.
- **Call waiting** in the phone is required for initial consultation, hold, and conferencing.
- **Conference Call** will be presented if the subscriber is provisioned with service. However, the registering SIP device must have Large Conference configured.

2.6.3.6 Service Permitted Usage Notes

The servicesPermitted parameter type is a list of bitmaps where each bit represents a service that can be applied to a connection. When a bit is set, the corresponding service is permitted. The following is the list of bitmaps (multiple bits may be set in this parameter):

- Call control services - the call control services supported by OpenScape Voice.
- Call associated services - the call associated services supported by OpenScape Voice.

1. This parameter indicates which of a subset of CSTA services are permitted.
2. When the servicesPermitted parameter is provided in an event, it applies to the connection at the monitored device. This may or may not be the same as the subject device.
3. If there are multiple connections at a device, the information reported in the servicesPermitted parameter may not accurately reflect all possible service restrictions and interactions between multiple connections at a device.
4. There may be situations in a switching function that cause a service to fail after being presented as permitted in the servicesPermitted parameter. This may be due to dynamic system and/or resource conditions that may cause service availability restrictions. The switching function shall provide the appropriate error code in the negative acknowledgement to the failed service request.

Note that all of the above give a good general view of what services permitted values can be expected under certain conditions. However, the CSTA application should make no assumptions based upon this information. The key point is for the application to be switch-type agnostic and use the actual servicesPermitted to determine what services can and cannot be issued at any given point in time.

2.7 OpenScape Voice Feature Interaction with CSTA and ONS

CSTA call control event flows are subject to interactions with other OpenScape Voice features. The services provided by CSTA and One Number Service (ONS) are included in the OpenScape Voice Feature Interaction Matrix (available in eDoku). The OpenScape Voice Feature Interaction Matrix provides feature-by-feature interactions for CSTA and ONS.

Note: One Number Service (ONS) is internal to Unify and is used over the CSTA interface between OpenScape Voice and OpenScape UC Application (for further details please refer to Appendix F, “One Number Service (ONS)”).

---

3 Administration and Management Considerations for CSTA

This section describes aspects of OpenScape Voice administration and Unify SIP phone configuration prerequisites to properly enable the CSTA services. Detailed administration and maintenance of OpenScape Voice, SIP phones and associated CSTA applications are described in the associated administration user and training documentation for each product.

3.1 Application Connectivity

3.1.1 OpenScape Voice—CSTA over TCP Connections

Application connectivity is managed using the CLI interface, which administers the CSTA protocol data associated with the CSTA Signaling Manager (SM).

- Maximum CSTA Application Sessions (1-256, default = 64)
- Maximum Monitored Endpoints (1 – 100,000, default = 100,000)
- CSTA Manager Virtual IP 1 (1.x.x.x - 255.x.x.x, where each “x” is an integer within the range 0 - 255. default = 0.0.0.0). Address used to establish TCP transport by CSTA application with the first CSTA SM instance.
- CSTA Manager Virtual IP 2 (1.x.x.x - 255.x.x.x, where each “x” is an integer within the range 0 - 255. default = 0.0.0.0). Address used to establish TCP transport by CSTA application with the second CSTA SM instance.
- Connect Port (1025 – 32000, default 1040). CSTA SM listening port.

Refer to Section 4.3.1.2, “Monitor Start” for a statement on maximum number of active monitors per connection.

Refer to *OpenScape Voice Config and Admin Using App-Level Management* for your installation for specific management requirements.

3.1.2 CSTA Application Connectivity to OpenScape Voice V9 Duplex Cluster

OpenScape Voice V9 exposes two CSTA TCP signaling manager IP address (i.e., cstasm1_vip and cstasm2_vip).

The default CSTA over TCP non-secure listener port is 1040.

The default CSTA over HTTP / SOAP non-secure listener port is 7780 but may be changed to 80 or 8080 by the system administrator.

Note: CSTA over TCP or HTTP / SOAP does not support secure access.

CSTA applications should either pre-configure cstasm1_vip and cstasm2_vip addresses or it may discover the IP addresses via a Domain Name Server (DNS SRV Records).

Regardless of which address the application connects to OSV the CSTA service is provided through a single process that is available through both virtual IP addresses.

3.1.2.1 Co-Located Node Cluster Failover

It is highly recommended that applications attempt to connect to cstasm1_vip first as shown in the Figure 5 below. If an application is unable to connect to cstasm1_vip (for example Network problems) then it should try to connect to cstasm2_vip.

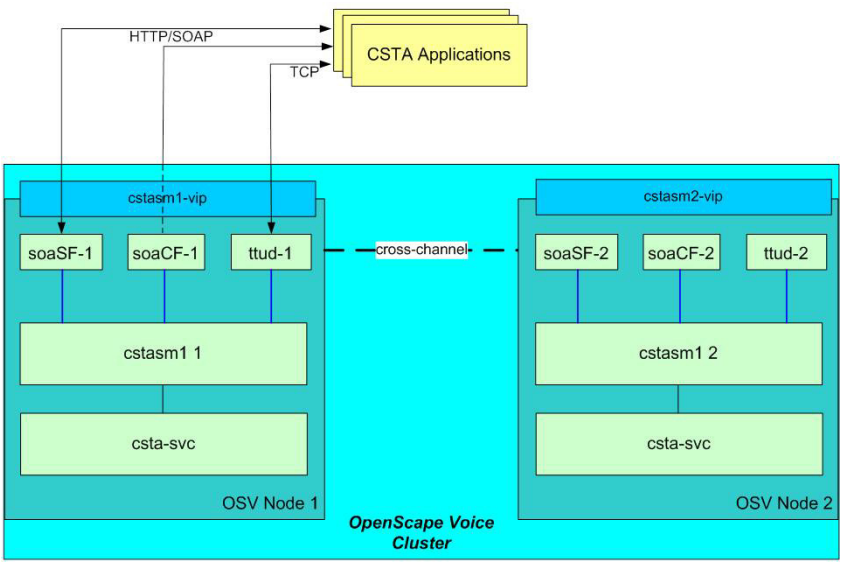


Figure 5 Co-Located Node Cluster

If the OpenScape Voice V9 node cluster is co-located (i.e., within the same network domain) and node1 is down then cstasm1_vip is migrated to node 2 as shown in the Figure 6 below. CSTA services are handled by cstasm1 on Node 2.

It should be noted that the migration of cstasm1_vip causes a disconnection of the TCP link. The application is expected to re-establish the connection. The logical location of cstasm1_vip is transparent to the application.

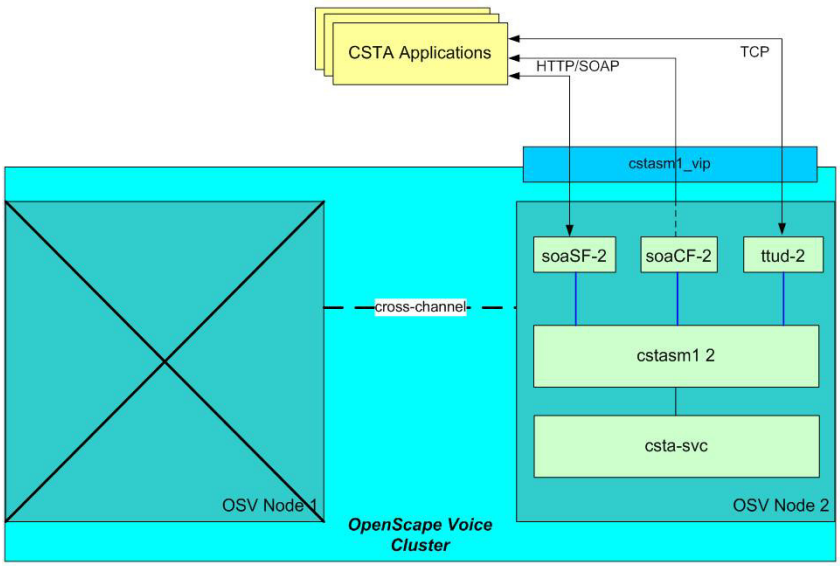


Figure 6 Co-Located Node Cluster Failover

3.1.2.2 Geographically Separated Node Cluster Failover

If node1 and 2 are geographically separated (IP addresses are in different network domains) and node1 is down, then only cstasm2_vip is available. It is important to note that in this configuration cstasm1_vip is not migrated to node 2 and cstasm2_vip must be used by the application.

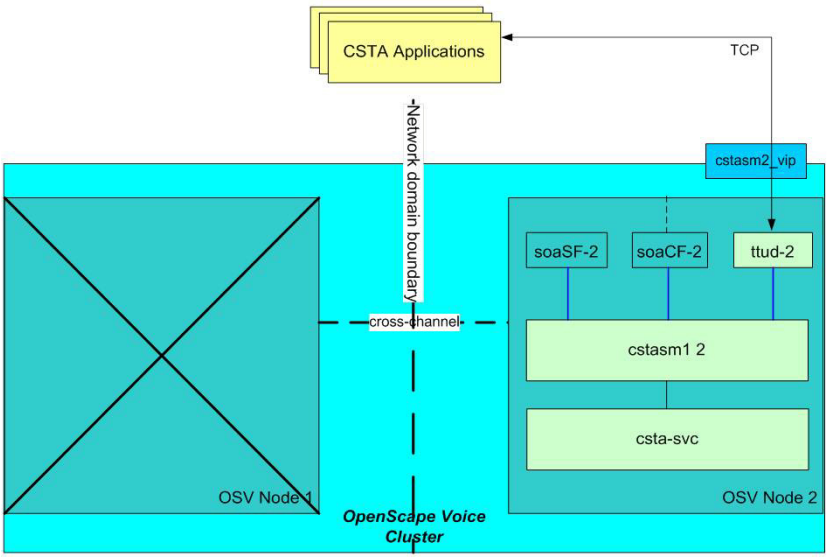


Figure 7 Geographically Separated Node Cluster Failover

It is important to note the following for geo-separated nodes:

- These nodes may be configured to operate independently in the event the cross-channel communications is not available. This situation is known as standalone or “split-brain” mode of operation. SIP devices attempt to register with an available node. For CSTA applications this means that in order to monitor all devices a connection to both nodes may be required. When CSTA detects standalone operation it notifies the application by sending a System Status service with Status = Partially Disabled. This CSTA system status remains in effect until the cross-channel communications is restored and the system returns to normal operation. Unless the CSTA application has been designed to operate with both OpenScape Voice V9 nodes simultaneously in standalone mode then some device services will be limited.
- CSTA SOAP signaling manager is currently designed to support only co-located systems. This is currently determined by the system RTP parameter Srx/Main/NodeSeparation. If this parameter is set to true then the CSTA SOAP signaling manager stays in standby mode.

3.1.2.3 Effects on CSTA Call Data on Node Failover

Active call media paths are always maintained after a failover. However, call data is only available to the recovering node if both nodes have the same release level. If the nodes are of different release levels, the CSTA call data for calls active during the failover shall not be available.

If both nodes have the same release level, OpenScape Voice V9 is able to maintain call related data for calls that were established at the time of failover and those that occured before the application re-established its connection. Applications that re-establish their connection are able to rediscover any active calls via CSTA snapshot services.

3.1.2.4 Connection Failure due to Network Problems

If the connection to cstasm1_vip fails due to network problems, then the application should attempt to connect cstasm2_vip as shown in the figure below. Please note that cstasm1 on node 1 handles CSTA services and events.

Attention: In this case, messages are sent over the clusters cross-channel which means that the overall call processing performance shall degrade. Therefore, it is highly recommended that if the application does connect to cstasm2_vip that it also periodically check for the availability of cstasm1_vip and if possible re-establish the connection to cstasm1_vip and tear down the connection to cstasm2_vip.

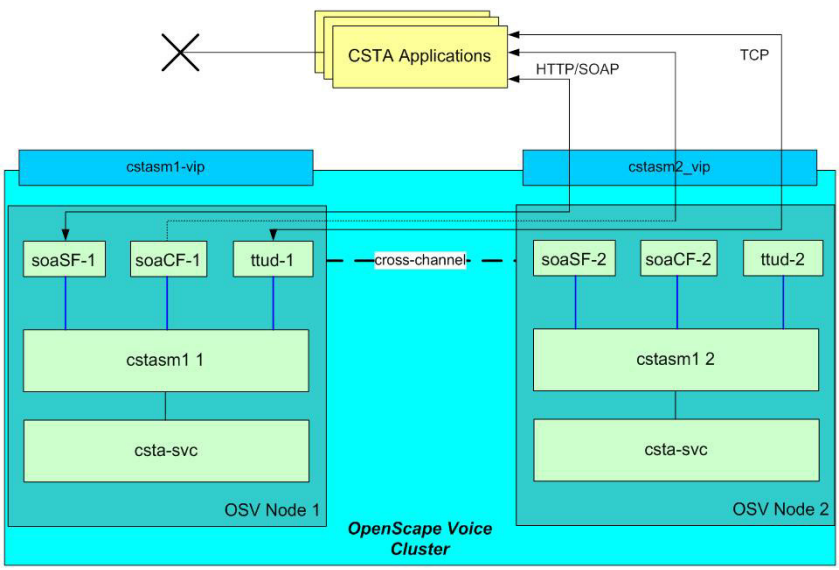


Figure 8 Connection Failure due to Network Problems

3.2 Subscriber Management Configuration (OpenScape Voice Assistant) or Expert Mode using CLI

3.2.1 BGL Services Affecting CSTA Services

Table 12 identifies some typical subscriber services that affect CSTA services.

Subscriber Service	Comments
Automatic Recall	Needed for recall flows
Call Forward Busy	Get / Set Forwarding
Call Forward No Answer	Get / Set Forwarding
Do Not Disturb	Get / Set DND for subscriber device
Call Forward Remote Activation	Call Forward Remote Activation
Call Forward Variable	Get / Set Forwarding
Call Forward Voice Mail	Single Step Transfer to Voice mailbox
Call Hold	Hold and Retrieve Call
Call Pickup Group	Call Pickup Group
Call Transfer	Consultation and compound call services
Call Waiting	Consultation and compound call services
Caller ID	Number Presentation services
CSTA	Enabler for all CSTA Services
ONS Inbound and Outbound	Enabler for ONS
Large Conference Service	Conference

Table 12 Relationship of OpenScape Voice Subscriber Services to CSTA Services

3.2.2 BGL with Registering or Non-Registering SIP Device

- CSTA Service must be active
- CSTA device capability category must be selected. Section 2.4, “Supported CSTA Services and Events (by Device Type)” describes how each device category listed below affects CSTA services:
  - Normal – for any SIP device
  - Type1 – for OptiPoint and OptiClient
  - CSTA over SIP – for OpenStage
- Provision SIP endpoint data. This is required whether a device registers for the BGL or not.

- It is recommended that Call Forward – Dependable be provisioned in the event a call is delivered to a currently unregistered SIP device.

3.2.2.1 Type1 and CSTA over SIP Device Type configuration

The following table identifies SIP phone/client configuration data required to support some CSTA services.

CSTA Service	Type1	CSTA over SIP	SIP Endpoint Configuration*	Comments
Agent State – Ready / Not Ready	Yes	Yes	Configure Hunt Make Busy feature toggle key on device.	Required for phone to provide the user interface for Agent State Ready / Not Ready indication.
Alternate Call	Yes	Yes	Auto reconnect = On	
Answer Call	Yes	Yes	Auto Answer=On	
Consultation Call	Yes	Yes	Auto Answer=On	
Conference	Yes	Yes	Auto Reconnect = On	To add a party to an existing conference.
Get / Set Call Forwarding	No	Yes	Centralized Feature Support = On Enable uaCSTA = On	Phone requires support of OpenScape Voice centralized Forwarding using uaCSTA.
Get / Set DND	No	Yes	Centralized Feature Support = On Enable uaCSTA = On	Phone requires support of OpenScape Voice centralized DND services using uaCSTA.
Get Microphone Mute	No	Yes	Enable uaCSTA=On	
Get Speaker volume	No	Yes	Enable uaCSTA=On	
MakeCall (hands free)	Yes	Yes	Auto answer = On	
Reconnect	Yes	Yes	Auto Answer = On Auto Reconnect = On	
Retrieve	Yes	Yes	Auto Answer = On Auto Reconnect = On	
Set Microphone Mute	No	Yes	Enable uaCSTA=On	
Set Speaker volume	No	Yes	Enable uaCSTA=On	
* Refer to OpenScape *Voice V9, Configuration, Administrator Documentation,* for information about configuring OptiPoint and OpenStage telephones.				

3.2.3 MLHG Pilot (Group Device / ACD) and Agent BGLs

Hunt Groups are used by applications to temporarily queue calls for distribution to available agents.

Refer to Appendix E, “MLHG and CSTA Capabilities” for a description of Hunt Group attributes and how these maybe used by CSTA-enabled applications.

Figure 9 illustrates an example of application configuration in a hosted environment to monitor and control hunt group call distribution.

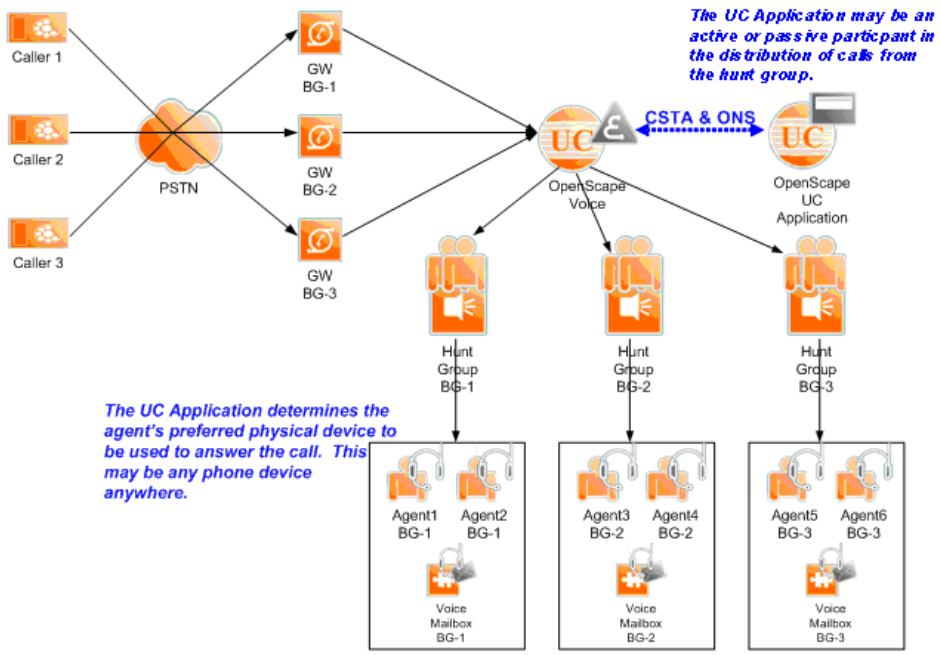


Figure 9 *Example: OpenScape Voice Hunt Group and OpenScape UC Application*

3.2.4 One Number Services (ONS) / ONS Subscribers

Attention: This section is for Unify internal use only. Refer to Appendix F, “One Number Service (ONS)” for further details.

ONS is provisioned under the User Mobility Services category. OpenScape Voice One Number Service requires CSTA subscriber service provisioning.

- **One Number Service - Inbound and Outbound (ONS-IO)**

  ONS-IO is a OpenScape Voice mobility service that enables CSTA and ONS aware applications to route, track, and control inbound and outbound calls using existing CSTA OpenScape Voice call control services for any device located anywhere.

3.2.5 SIP Multiple Contact Registration and Feature Interaction

OpenScape Voice supports multiple SIP contacts registration for the same address of record. Thus SIP messages sent to the address of record from the system are forked to multiple devices.

To prevent conflicts with feature interactions OpenScape Voice restricts service initiation by multiple contacts to only the SIP contact ID already active on a call.

Refer to individual CSTA services for operational notes regarding interaction with multiple contacts.

The CSTA call control event flow involving multiple contacts with the same deviceID has been enhanced to ensure that duplicate connectionIDs are not presented on the CSTA device monitor.

3.2.6 Shared Multiple Appearance (Keyset Operation)

While a keyset device may be provisioned for CSTA services, the CSTA interface does not provide a keyset model. All services are provided for the keysets prime line.

OpenScape Voice does not currently support a full keyset model as defined by ECMA-269 Annex A. However, OpenScape Voice does provide the following functionality for devices configured as keysets:

- A monitored DN (prime line or Address of Record) that appears as a line appearance (non-prime line) on another device is selected (reserved), CSTA flows a Services Initiated for the monitored DN. The initiatingDevice parameter is set to the prime line of the device where the line is reserved or in-use.
- CSTA provides normal event flows for any monitored DN used in a keyset arrangement.
- CSTA call control and associated call control services are blocked (via servicesPermitted) while a monitored DN is in-use as a secondary appearance on another device. All service requests are rejected.
- A privateData tag named “keyOperation” is included in call control events presented on the primary line indicating that the primary line device is not involved in the call. Applications should use this information to appropriately manage call control user interface associated with the user’s primary line.
- The deviceID SFR format has been enhanced to include a “keyID” tag that identifies the shared line device that is currently involved in an active call.
- CSTA non-call related CSTA services are supported while a monitored DN is in-use as a secondary appearance on another device.
- DeviceID format for subscriber DNs configured as Keysets with Line or Device based operation the following presentation rules apply:
  - **Line Based Keyset Operation**

    The DN field and the displayNumber parameter of the SFR deviceID are the same number (that is, the line in use).
  - **Device Based Keyset Operation**

    When an originating call is answered, the DN is the line in use; the displayNumber is the display number of the device in use. Otherwise the DN and the displayNumber portion of the SFR are the same number (that is, the line used).

    For terminating calls to device based line appearances on a keyset device, the called device is the line that is called and the displayNumber is the display number for that line. Upon answer of the call by a device

based line appearance, the called device is the line that answered, but the displayNumber is the display number for the device upon which the line appears.

Exception: If the CISNUM service is not enabled at the caller's device, or if the caller is in another BG, such that called party display information is not presented to the phone, CSTA attempts to provide display information using NDAL API. In this special case, the displayNumber is always the display number for the line that was called.

3.2.7 Multiple Contacts (SIP Forking)

CSTA monitors a provisioned subscriber's address of record (AoR). If multiple bindings are in a call independently, each binding will be treated as a separate call. CSTA maintains no association between two calls at the AoR. The uniqueness of each binding is maintained via the UCE context-id of the call (callID), IP Address and Port associated with the binding.

CSTA Services that act on a call shall use the call-id associated with the call to invoke services. This will result in only the particular binding being acted upon for a specific CSTA Service. In the case of a CSTA Consult Service Request, the CSTA Manager will ensure that the associated device used for initiating the A-side of the consultation leg will be the one that is already in a call identified in the Consult Service Request connection-id parameter. The other bindings will not be considered for consultation service request.

The CSTA Service and CSTA Manager shall ensure that only one Special considerations for ONS subscribers: and only one Out of Service event is sent to the CSTA applications per DN.

3.2.8 Global Number Format (GNF) Replacement - RTP Parameter

The GNFPrefixReplacement RTP parameter and Prefix Access Code (PAC) table must be correctly provisioned to ensure proper translation of deviceIDs sent by an application in Global Number Format. Any service requests sent by the application with GNF in any portion of a deviceID (including OND) is rejected with an InvalidDestination error code.

GNFPrefixReplacement—Unique prefix access code string of up to 15 characters (allowed: 0-9, *, #). CSTA SM replaces the '+' sign with this unique prefix access code string and the resulting number is translated as NoA = 'unknown'. The administrator must provide the correct entries in the Prefix Access Code (PAC) table to support the translation of the unique prefix access code to breakout to a common numbering plan of the Destination Code Table.

OpenScape Voice V9 number translation now supports "+" in the number plan table. The GNFPrefixReplacement will be automatically set to allow GNF in CSTA service request as a default.

3.3 Failure and Restart Considerations

3.3.1 Simplex OpenScape Voice System

If the CSTA service is restarted for any reason TCP link is dropped and must be reestablished by the application. Active monitors is stopped unless they are started under an application session as defined in Section 4.5, “Application Session Services”.

System status events provide information relating to restart and recover. Refer to Section 4.2.2.2, “System Status”for services provided in the event of a restart of the CSTA service.

3.3.2 Duplex OpenScape Voice System (co-located)

Same as Simplex except that after node switch over the CSTA TCP IP address is maintained and migrated to the backup OpenScape Voice node.

3.3.3 Duplex OpenScape Voice System (geo/network separated)

Section 3.1.2.2, “Geographically Separated Node Cluster Failover” describes how CSTA TCP IP addresses are handled in this configuration.

3.3.3.1 CSTA Application using Cold Stand-By Operation

An instance of a CSTA application is deployed in each subnet. One CSTA application is configured in OpenScape Voice and used during normal operation. The other CSTA application is a cold standby application. Once, the server of the configured CSTA application is out of order, OpenScape Voice is reconfigured to use the server with the cold standby CSTA application.

CSTA applications connect to the CSTA signaling manager using a single IP address. Today this IP address floats from the primary node to the secondary node upon node failure, forcing all traffic through the fiber-link.This requires manual invocation of cold stand-by application server.

3.3.3.2 CSTA Application using Hot Stand by Operation

OpenScape Voice also supports DNS. Ideally, CSTA applications should also support the DNS SRV record and actually look for a live CSTA signaling manager in a list of IP addresses provided. However, if this is an option that application may consider. If a hot standby capability is not possible due to time/effort then a cold stand-by should be used.

3.4 ECMA 354 Application Session Services

It is highly recommended that applications support ECMA 354 Application Session Services. Application Session Services enable OpenScape Voice and the application to preserve established device monitors for a negotiated period of time.

Application session services also provide a mechanism to detect application heartbeat failure for MLHG backup call distribution.

Refer to Section 4.5, “Application Session Services” for service details.

3.5 Call Admission Control (CAC) Subscriber Re-routing

OpenScape Voice supports subscriber re-routing over the PSTN in case with Resource Management failure or WAN/MAN failure. As a result of subscriber re-routing CSTA call flows are need to be understood by the application. Detailed CAC Resource Management and WAN failure call flows are defined in Appendix A, “CSTA Call Scenarios”.

3.6 CSTA Application Management

- OSV management enables the display of currently connected / provisioned applications.
- Application IP addresses may be added or deleted.
- Packet filter rules are automatically created or removed when applications are added or deleted.
- There are several application interface options that have been defined over the releases to enable or disable presentation of privateData or other new functionality. This enables OSV to be upgraded without affecting backward compatibility of installed applications.
- The following interface options are currently supported:
  - Enable Consultation Call for connections on hold

    When this option is enabled (checked) Consultation Call is presented in the servicesPermitted for a call on hold when this option is checked. This feature is disabled (unchecked) by default.
  - Enable Private Data Presentation - V8

    When this option is enabled (checked) all privateData elements introduced through Release 8 shall be presented. This option is disabled (unchecked) by default.

- Refer to OSV Administration Documentation for more details on application management.

4 OpenScape Voice CSTA Service Description

Global Service Description Statements:

- Supported negative service response error values are described in Section 2.6.2, “Negative Service Response”.
- Unless otherwise specified the OpenScape Voice operational model should not differ from related ECMA specification.

4.1 Capabilities Exchange

4.1.1 Services

4.1.1.1 Get CSTA Features

The Get CSTA Features service obtains the list of CSTA services and events supported by the switching function.

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Service Request Parameters

No required parameters.

Service Response Parameters

This service follows the atomic acknowledgement model for this request.

Parameter Name	Supported	Comments
supportedServices	Y	Specifies the list of CSTA services supported by the switching function. Each entry in the bitmap represents supported services.
supportedEvents	Y	Specifies the list of CSTA events supported by the switching function. Each entry in the bitmap represents a supported event.

OpenScape Voice Operational Notes

- This service provides a static view of services and events defined in Section 2.4, “Supported CSTA Services and Events (by Device Type)”.
- Rules for dynamic feature presentation refer to Section 2.6.2, “Negative Service Response”.

4.1.1.2 Get Logical Device Information

The Get Logical Device Information service is used to obtain the current set of characteristics / capabilities associated with the logical element of a given device.

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Service Request Parameters

Parameter Name	Content	Supported	Comments
device	DeviceID	Y	Specifies the device being queried. **Note:** Network Interface Devices must be present in Device Number (DeN) format (see Section 2.6.1.3, “Device Number (DeN) Format”)

Service Response Parameters

This service follows the atomic acknowledgement model for this request.

Parameter Name	Supported	Comments
deviceCategory	Y	Specifies the device category (station, ACD device, and so on) of the device in the service request. Supported values: • Station (Default) – Any subscriber DN • Group (Hunt) when deviceID is OpenScape Voice MLHG Pilot DN. • Network Interface Device Number
groupDeviceAttributes	Y	Specifies the group device attributes of the device being queried. Bitmap values supported is: • Hunt - when deviceCategory is set to “Group”
hasPhysicalElement	Y	Specifies if the device has a physical element associated with this device identifier. • FALSE - The device does not have a physical element (for example, Group device) • TRUE - The device does have a physical element.
acdModels	Y	Specifies the type of ACD Model(s) that are present at this device. If a bit is TRUE, then the specified model is supported. The following is the list of bits (multiple bits may be set): • Visible ACD-related Devices - FALSE • Non-Visible ACD-related Devices - FALSE OpenScape Voice always sets both to zero (false) since OpenScape Voice does not support ACD devices.

Parameter Name	Supported	Comments
appearanceAddressable	Y	Specifies whether the appearances of the logical element are addressable (using the Call Appearance "CA" string or the physical element extension "EXT" string in the Switching Function Representation Device Identifier format). This is Sent when device is a keyset. The only supported value is: • FALSE - Keyset appearances are not addressable.
appearanceType	Y	Specifies the type of appearances associated with the logical element. Supported value(s): • Selected-Standard
appearanceList	Y	Specifies the list of device identifiers that are available at the logical element. The parameter is mandatory if the appearances are addressable and if it is Selected Standard or Basic Standard Type. OpenScape Voice usage for MLHG pilot is non-standard for Group deviceCategory: • When the logical device is an MLHG Pilot DN a list contains the member devices provisioned in the MLHG. This list only contain appearances DNs that can be observed and/or controlled within the OpenScape Voice domain. • When the logical device is a station provisioned as a keyset the list contains an entry for each line appearance allowed for this subscriber. The string for each entry has the following format: – AppearanceDN – : (colon) – Originating Mode (0=Device Based, 1=Line Based) – , (comma) – Terminating Mode (0=Device Based, 1=Line Based)
associatedGroupList	Y	This parameter specifies the list of device identifiers for all the other devices which are members of this group deviceID. This optional list shall only contain devices within the switching sub-domain that may be either observed and / or controlled within the switching subdomain. The following conditions are supported: • When logical device ID is an agent (member of MLHG) then this list contains the MLHG Pilots the agent is associated with.
mediaClassSupport	Y	Specifies the media class of calls that the device can support. Support values: • Voice • Image Refer to Section 4.1.1.3, "Get Switching Function Capabilities" for services and events that support media characteristics.

Parameter Name	Supported	Comments
logDevServList	Y	Specifies a list of capability bitmap parameter types corresponding to categories of services. Each bitmap entry in the lists represents a service that applies to a logical device that is supported by the device. This includes but is not limited to the following categories of services: • callControlServList • callAssociatedServList • callAssociatedServList • logicalServList Refer to Appendix C, “Supported CSTA Services and Events by Device Type” for supported CSTA services by device type.
logDevEvtsList	Y	Specifies a list of capability bitmap parameter types corresponding to categories of events. Each bitmap entry in the lists represents an event that applies to a logical device that is supported by the device. This includes but is not limited to the following categories of events: • callControlEvtsList • callAssociatedEvtsList • logicalEvtsList Refer to Appendix C, “Supported CSTA Services and Events by Device Type” for supported CSTA events.
deviceMaintEvtsList	Y	Specifies a list of bitmaps. Each bitmap entry represents a device maintenance event that is supported by the device. Supported values: • Back in Service • Out of Service
maxDevicesInConf	Y	Specifies the maximum number of devices both within and outside the switching function that this device may conference into a call. The minimum value that can be supplied for this value is 3. • If device uses Large Conference services this value range is 3-48 members.
transAndConfSetup	Y	Specifies the different ways that this device can set up for a conference and/or transfer. OpenScape Voice supports: • Consultation Call

Parameter Name	Supported	Comments
privateData	Y	This services returns the following private data elements: • **devType** with the possible values: – SIP – MGCP – MLHGPILOT **Note:** This parameter is not provided for other device types. • **busGroupID** with the following: – Business Group ID **Note:** This parameter is not provided if subscriber is not member of BG. • timeZoneLocation - Default is “Local” meaning switch time zone. Refer to OpenScape Voice Assistant for supported time zone values. • **cidSuppressionActive**; boolean. “true” means CID suppression is active. “false” means not active. • **altCidList** with the following sub-elements: • altCid – repeated for each entry – altIndex – integer; identifies the alternate CID index number (0—09) – altDigits – string; alternate calling device number presented to the called device – altName – string; alternate calling device name presented to the called device – altPresentName boolean; “true” means name should be presented. “false” means name should not be presented. If the queried device is a Network Interface Device (NID) the following privateData elements shall be included. • endPointProfileName – The associated Endpoint Profile Name is provided if the queried device is an Network Interface. • endPointProfileNumberingPlan - The associate Endpoint Profile Numbering Plan is provided if the queried device is an Network Interface. • endPointName – The associate Endpoint Name is provided if the queried device is an Network Interface. • endPointType - The associate Endpoint Type is provided if the queried device is an Network Interface. The possible values are: – SIP Private Network – SIP Trunking – SIP-Q Signaling • nameProviderActive – True or False

OpenScape Voice Operational Notes

None

4.1.1.3 Get Switching Function Capabilities

The Get Switching Function Capabilities service is used by the computing function to obtain the current set of capabilities for the entire switching function.

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Service Request Options

No required parameters.

Service Response Parameters

This service follows the atomic acknowledgement model for this request.

Parameter Name	Support	Comments
switchingSubDomainName	Y	Specifies the name of switching sub-domain which distinguishes it from other switching sub-domains. Currently supported value: • OpenScape Voice (Enterprise)
manufacturerName	Y	Specifies the name of the manufacturer of the switching sub-domain. Currently supported value: Unify
profiles	Y	Specifies the CSTA Profiles supported by the switching function. Currently supported profiles: • basicTelephonyProfile
deviceIDFormat	Y	Specifies the types of device ID formats supported by the switching function in service requests. If a bit is TRUE, then the specified format is used by the switching function. In addition to dialable digits 0-9, the following is the list of the possible formats (multiple bits may be set): • Asterisk “*” • Hash “#” • SF Representation format - “NM” SF Representation format – Generic **Note:** Some additional formats may be reported as true but are not officially supported.

Parameter Name	Support	Comments
swdomainFeature	Y	Specifies which features are supported by the switching function. Supported values: • Forward – isForwardingBefore • Forward Default – swFunctionDefaultSettings • Forward Default – userSpecific • Connection Failure – Neg. Ack. • Connection Failure – supportFailedWithAssConn • Other – External Calls Incoming Calls • Other – External Calls Outgoing Calls • Other – recall • Other – callBack • Other – prompting
swAppearanceAddressability	N	Specifies what types of appearance addressability is available within the switching sub-domain.
swAppearanceType	Y	Specifies what types of appearances are available within the switching sub-domain All values are set to False.
ignoreUnsupportedParameters	Y	Specifies how the switching function handles unsupported optional parameters in service requests. Supported values: • Ignore parameters - This indicates that the switching function treats unsupported optional parameters as if they were not present.
mediaClassSupport	Y	Specifies the media class of calls that the device can support. Supported values: • Voice • Image

<table>
<tr><th>Parameter Name</th><th>Support</th><th>Comments</th></tr>
<tr><td>callCharacteristicsSupported</td><td>Y</td><td>Specifies the characteristics that the switching function sends in CSTA events:

Parameter — Supported
acdCall — Y
lowPriorityCall — N
priorityCall — Y
highPriorityCall — Y
maintenanceCall — N
directAgent — Y*
assistCall — N
voiceUnitCall — N
privateCall — N
personalCall — N
sensitiveCall — Y
confidentialCall — N
encryptedCall — Y**

Note: OpenScape Voice does not support callCharacteristics parameter in service requests. The following callCharacteristics are provided in Call Control and Call Associated Feature events.

Note: Conference Encrypted call is also supported.</td></tr>
<tr><td>privateDataFormat</td><td>Y</td><td>Specifies the format(s) of the privateData information supported by the switching function. Supported value:
• octetStringFormSF

Note: This parameter is not sent. However, privateData is supported in some services and events as indicated in the subsequent sections.</td></tr>
<tr><td>dynamicFeatureSupported</td><td>Y</td><td>Specifies how the switching function provides the servicesPermitted parameter on events. Supported value:
All – for all supported events.

Note: This parameter is not sent. However, OpenScape Voice does support and provide dynamic feature presentation as described throughout this specification.</td></tr>
</table>

Parameter Name	Support	Comments
deviceOnDeviceMonitorFilter	Y	Specifies the complete monitorFilter parameter that is supported by the switching function when the monitorObject is a device and the monitorType is a device. Supported values: • Callcontrol - Filter is supported for each supported call control event in Section 2.3.5, “Application Link Security Considerations”. • Logical device - Filter is supported for each supported call control event in Section 4.11, “Logical Device Feature Services”.
connectionView	Y	Specifies the meaning of the primary and secondary old call parameters in the Conferenced and Transferred events. Supported values: • local view
maxLengthParameters	Y	Each value is the switching function’s maximum length (in octets/ characters) for the corresponding parameters and parameter types. The computing function should not send larger data or the service request is rejected. The following list provides the different parameters and parameter types for which a maximum value is provided. The number in parenthesis specifies the maximum possible length. Supported parameters (max. length) • agentID (32) • callID in the ConnectionID (16) • CSTAPrivateData parameter type: (128) • Device Identifiers (128) • userData(256) • charactersToSend parameter: (32) If any of the above values is zero, then the parameter or parameter type is not supported.
servEvtsList	Y	Specifies a list of capability bitmap parameter types corresponding to categories of services. This list corresponds to those services and events listed in Section 2.4, “Supported CSTA Services and Events (by Device Type)”. Supported parameters: • capExchangeServList • systemStatServList • monitoringServList • snapshotServList • callControlServList • callControlEvtsList • callAssociatedServList • callAssociatedEvtsList • logicalServList • logicalEvtsList • deviceMaintEvtsList If a list entry is not included in the list, then the corresponding category of services/events is not supported by the switching function.

Parameter Name	Support	Comments
maxLengthParametersContinued	Y	Each value is the switching function's maximum length (in octets/ characters) for the corresponding parameters and parameter types. The computing function should not send larger data or the service request will be rejected. The following list provides the different parameters and parameter types for which a maximum value is provided. The number in parenthesis specifies the maximum possible length. Supported parameters (max. length): • callQulifyingData (324 bytes)
* The description of this callCharacteristic was modified in ECMA 269 Edition 7. The value "DirectAgent" is used to indicate a direct call that over-rides the destination re-direction settings for example DSS operations. ** This is a new callCharacteristic defined in ECMA 269 Edition 7.		

OpenScape Voice Operational Notes

None

4.1.1.4 Get Physical Device Information

Service Not Supported

The Get Physical Device Information service is used to obtain the current set of characteristics / capabilities associated with the physical element of a given device.

Note: OpenStage devices support some physical device capabilities such as, volume and mute control. Refer to Section 4.10, "Physical Device Features" for details.

4.1.1.5 Get Switching Function Devices

The Get Switching Function Devices service is used by the computing function to obtain the current set of devices in the application working domain along with their associated device categories and associated device names.

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, "Negative Service Response".

Service Request Options

Parameter Name	Supported	Comments
requestedDeviceID	Y	Specifies the device identifier of the device being queried. If the device ID is not provided then a list of all supported device category. **Note:** Network Interface Devices must be present in Device Number (DeN) format (see Section 2.6.1.3, "Device Number (DeN) Format")
requestedDeviceCategory	Y	Specifies that only devices of the requested category be provided. Supported categories: • Group (Hunt) – for MLHG Pilots • Network Device

Service Response Parameters

This service follows the multi-step acknowledgement model for this service request.

Positive Acknowledgment

The positive acknowledgement for the Get Switching Function Devices service indicates that one or more Switching Function Devices services will subsequently be generated by the switching function.

Parameter Name	Supported	Comments
serviceCrossRefID	Y	Specifies the correlator used to associate subsequent Switching Function Devices services to this service request.

OpenScape Voice Operational Note

MLHG and Network Devices are the only device categories supported.

4.1.1.6 Switching Function Devices

The Switching Function Devices service is used by the switching function to provide a list of devices in the application working domain. This service is generated as a result of the Get Switching Function Devices service.

The switching function may generate a sequence of Switching Function Devices services, individually referred to as segments, in response to a single Get Switching Function Devices service request.

Acknowledgement Model

There is no positive acknowledgement. This is sent as a result of Get Switching Function Devices service request.

Service Request Parameters

Parameter Name	Supported	Comments
serviceCrossRefID	Y	Specifies the cross reference used to associate the Switching Function Devices service request to the Get Switching Function Devices service request.
segmentID	Y	Specifies the segment number of this message. Each successive segment number in the sequence increments the segmentID by one.
lastSegment	Y	Specifies (True / False) if this segment is the last one associated with the serviceCrossRefID.
deviceList	Y	Specifies the list of device Identifiers representing the devices that can be controlled and/or observed. Supported parameters: • deviceID – specifies device ID for the device category Provides list of deviceID depending on the requestedDeviceCategory; Group or Network Devices. **Note:** NOTE: Network Interface devices are presented in Device Number (DeN) format (See Section 2.6.1.3, “Device Number (DeN) Format”)

Service Response Parameters

There is no service request completion conditions associated with this service

Positive Acknowledgement

There are no positive acknowledgements associated with this service request.

OpenScape Voice Operational Note

Supports Group (MLHG) and Network Interface (Gateway) Devices.

4.2 System Services

4.2.1 Registration Services

4.2.1.1 Change System Status Filter

Service Not Supported

The Change System Status Filter service is used by the computing function to change the filter options for a current system registration.

4.2.1.2 System Register

The System Register service is used by the computing function to register to receive system services from the switching function.

The computing function may be required to register for system services before it can receive any system service requests from the switching function.

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Service Request Parameters

Parameter Name	Supported	Comments
requestTypes	Y	Specifies the system services that are being registered. Supported parameters: • System Status²
requestedStatusFilter	Y	Specifies the requested set of System Status Types to be filtered (not sent) by the switching function. OpenScape Voice does not support status filtering and responds with actualStatusFilter accordingly.

Service Response Parameters

This service follows the atomic acknowledgement model for this service request.

Positive Acknowledgment

Parameter Name	Supported	Comments
sysStatRegisterID	Y	Specifies the system registration identifier for this registration.
actualStatusFilter	Y	Specifies the actual set of System Status Types that is filtered (not sent) by the switching function. OpenScape Voice responds with all statuses. No filtering is provided.

OpenScape Voice Operational Notes

- Only one system registration is supported per TCP link / Application Session. Subsequent System Register requests is negatively acknowledged.
- The registration ID is no longer valid if the TCP link fails or connectivity adaptor is restarted.

4.2.1.3 System Status Register Abort

Service Not Supported

The System Register Abort service is used by the switching function to asynchronously cancel an active system registration. This service invalidates a current systems status registration.

4.2.1.4 System Status Register Cancel

Service Not Supported

The System Register Cancel service is used to cancel a previous system registration. This request terminates the system registration and the computing function receives no further system service requests for that system registration once it receives the positive acknowledgement to the System Register Cancel request.

4.2.2 Services

4.2.2.1 Request System Status

The Request System Status service is used by the computing function to obtain (that is, query) the system status of its peer function.

Service Request Parameters

Parameter Name	Supported	Comments
sysStatRegisterID	Y	Specifies the system registration identifier for this registration.

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Service Response Parameters

This service follows the atomic acknowledgement model for this service request.

Positive Acknowledgement

Parameter Name	Supported	Comments	
systemStatus	Y	Specifies the status of the function issuing the service request.	
		Status	**Supported**
		• Initializing • Enabled • Messages Lost • Disabled • Partially Disabled • Overload Imminent • Overload Reached • Overload Relieved	Yes Yes No (Future) Yes No (Future) No (Future) Yes Yes

OpenScape Voice Operational Notes

None

4.2.2.2 System Status

The System Status service is used by the computing function or switching function to report its system status to its peer function. The indicated status may or may not have changed since the last System Status request was issued. This service can also be used to implement a heartbeat mechanism between the two functions. This service is bidirectional. The table below only reflects the OpenScape Voice system status event implementation. System status bidirectional scenarios are described in the following:

Table 4-1 - System Status Service from OpenScape Voice to Application

Table 4-2 - System Status Service from Application to OpenScape Voice

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Service Request Parameters

Parameter Name	Supported	Comments
sysStatRegisterID	Y	Specifies the system registration identifier associated with the system registration for this request.
systemStatus	Y	Specifies the status of the function issuing the service request.

Status	Supported
• Initializing	Yes
• Enabled	Yes
• Messages Lost	No (Future)
• Disabled	Yes
• Partially Disabled	Yes
• Overload Imminent	No (Future)
• Overload Reached	Yes
• Overload Relieved	Yes

monitorObjectList	C	Parameter sent when systemStatus = Messages Lost. The list contains all monitor

Service Response Parameters

This service follows the atomic acknowledgement model for this request.

OpenScape Voice Operational Notes

- OpenScape Voice V9 sends a System Status service immediately after an application establishes a TCP connection. The connected application sends a System Status response within 20 seconds or OpenScape Voice V9 will drop the TCP connection.
- System Status messages are sent after the System Register service request is received or after a System Status request from the computing function when used as a heartbeat mechanism.

- Currently OpenScape Voice does not provide a mechanism to enable periodic system status using systemStatusTimer in Get Switching Function Capabilities positive acknowledgement.
- A system status of partially disabled occurs when geographically separated OpenScape Voice V9 nodes are running in standalone mode.
- The following tables describe reasons for system status services requests.

OpenScape Voice -> Application or Application Platform		
systemStatus cause value	**Reasons or meanings for this status event**	**Expected Reaction**
Messages Lost	• OpenScape Voice detected that a CSTA event may have been lost during internal processing. • OpenScape Voice is restarting. In this case, service request messages are rejected until the OpenScape Voice link is re-established.	The application or application platform should take the necessary recovery actions to keep its databases updated while facing the prospect that at least one CSTA event may have been lost.
Partially Disabled	• OpenScape Voice geographically separated node is running in standalone mode. • In this mode some registering devices may be unavailable (out-of-service) until normal operation is re-established.	The application should be aware that some devices may be out-of-service in this node.

OpenScape Voice -> Application or Application Platform		
systemStatus cause value	**Reasons or meanings for this status event**	**Expected Reaction**
Disabled	• OpenScape Voice is shutting down, restarting or reloading. It is very possible that the "Disabled" event may not reach the application. • The CSTA service has been administratively disabled on OpenScape Voice.	The application should assume that OpenScape Voice is in a "not ready state and discontinue any further service requests until OpenScape Voice has signaled that it has again entered the ready state. **NOTES:** The Ready state is signaled by a subsequent System Status event with the Enabled cause value. • While in the Disabled state no service request is acknowledged by OpenScape Voice. • All active monitors are cleared and should be considered lost by the application. • If OpenScape Voice is shutting down the Enabled system status is not sent until OpenScape Voice has been restarted. • If OpenScape Voice is performing a restart it could take several minutes for the Enabled system status to be sent.
Enabled	• OpenScape Voice is ready • OpenScape Voice CSTA service has been administratively re-activated and ready for service requests from the application.	Begin sending service requests. Monitors are cleared. **Note:** If OpenScape Voice did not send an Initializing system status prior to the Enabled event then all previously established monitors remain valid.
Initializing	OpenScape Voice CSTA service is restarting and all monitors are no longer valid. OpenScape Voice always sends 1 mandatory Initialization system status prior to an application registration.	Same expectations as the Disabled system status.

OpenScape Voice -> Application or Application Platform		
systemStatus cause value	**Reasons or meanings for this status event**	**Expected Reaction**
Overload reached	OpenScape Voice is declaring that an overload condition has been reached.	When OpenScape Voice sends the systemStatus with the overloadReach cause value the application is blocked from sending new MakeCall, ConsultationCall and SingleStepTransfer service requests. This condition remains until OpenScape Voice sends the System Status = overloadRelieved.
Overload relieved	The previously declared Overload Reached condition has been relieved.	Return to normal processing and begin sending service requests.
Normal	Response to application status (that is, Loopback). Maybe used to provide a heartbeat. However, currently no periodic heartbeat is provided by OpenScape Voice.	If capable the application must positively acknowledge the system status event.

System Status message are sent after the application registration is received using the requestTypes System Status and maybe used by the application as a heartbeat mechanism.

Note: Currently OpenScape Voice does not provide a mechanism to enable periodic system status using systemStatusTimer in Get Switching Function Capabilities positive acknowledgement.

Application or Application Platform –> OpenScape Voice		
systemStatus cause value	**Reasons or meanings for this status event**	**Expected Reaction**
Enabled	Typically all active monitors have been cleared.	OpenScape Voice assumes that the application has cleared all active monitors if Application Session Service timer has expired or no application session has been started.
Normal	Response to application status (that is, Loopback) Maybe used to provide a heartbeat. However, currently no periodic heartbeat is expected by OpenScape Voice.	If capable, OpenScape Voice positively acknowledges the system status event.

4.2.2.3 Switching Function Capabilities Changed

Service Not Supported

The Switching Function Capabilities Changed service is used to indicate that switching function level capability information available using the Get Switching Function Capability service has changed.

4.2.2.4 Switching Function Device Changed

Service Not Supported

The Switching Function Devices Changed service is used to indicate that information associated with the current set of devices that can be controlled and observed in the switching subdomain has changed.

4.3 Monitoring Service

4.3.1 Services

4.3.1.1 Change Monitor Filter

The Change Monitor Filter service is used to modify the set of event reports that are filtered out (not sent) over an existing monitor. The new set events to be reported are listed in the service acknowledgement.

Service Request Parameters

Parameter Name	Supported	Comments
crossRefIdentifier	Y	This parameter indicates the monitor for which to change the filter.
requestedFilterList	Y	This parameter specifies the requested set of events to be filtered out (not sent) by the server. Refer to Appendix C, “Supported CSTA Services and Events by Device Type” for supported CSTA events.

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Service Response Parameters

This service follows the atomic acknowledgement model for this request.

Positive Acknowledgement

Parameter Name	Supported	Comments
actualFilterList	Y	This parameter specifies the requested set of events to be filtered out (not sent) by the server. The actual events filtered may be different from the requested filter. If the actual and requested filters are the same this parameter is not be sent in the positive response. Refer to Section 2.4 for supported CSTA events.

OpenScape Voice Operational Notes

None

4.3.1.2 Monitor Start

The Monitor Start service initiates event reports (otherwise known as events) for a device.

OpenScape Voice does not support call monitoring.

Service Request Parameters

Parameter Name	Supported	Comments
monitorObject	Y	Specifies the monitor object of a device to be monitored. Supported option: • Device (Device ID) **Note:** Network Interface Devices must be present in Device Number (DeN) format (see Section 2.6.1.3, “Device Number (DeN) Format”) **Note:** A deviceID of "BBBB" can be used to initiate a bulk monitor of all ONS-enabled subscribers.
requestedMonitorFilter	Y	This parameter specifies the requested set of events to be filtered out (not sent) by the switching function. It is a bitmap of all events defined in this Standard. Refer to Appendix C, “Supported CSTA Services and Events by Device Type” for supported CSTA events supported by the monitorFilter. **Note:** CSTA limits the events on NID to Offered, Originated, Connection Cleared and Call Information; however the application may filter these events. If they are filtered CSTA SM shall honor the filter request.
monitorType	Y	Specifies the type of monitor requested. Supported options: • device-type (default if not provided)
requestMonitorMediaClass	N	Specifies the media classes (voice, digital data, Email, message, etc.) of calls that are being requested to be monitored for the monitorObject.

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2.

Service Response Parameters

This service follows the atomic acknowledgement model for this request.

Positive Acknowledgement

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Specifies the monitor object of a device to be monitored. Supported options: • Device (Device ID) **Note:** In the case of a bulk monitor the deviceID returned by OSV is "XR:BBBBx". The "x" is an ordinal number that makes the bulk monitor object unique. The application needs to understand that all CSTA event presented when bulk monitoring will be presented as the subscriber's serviceID of the subscriber. When an CSTA event is generated for an ONS-enabled subscriber and bulk monitoring is enabled CSTA SM presents the monitorCrossRefID parameter equal to the subscriber's provisioned serviceID as shown below: <monitorCrossRefID>15619231439</monitorCrossRefID> However, if the CSTA event is generated for a WebRTC-enabled subscriber CSTA SM will append the string ":webRTC" to the serviceID. In this case the monitorCrossRefID shall be presented as follows: <monitorCrossRefID>15619231439:webRCT</monitorCrossRefID>
requestedMonitorFilter	Y	This parameter specifies the requested set of events to be filtered out (not sent) by the switching function. It is a bitmap of all events defined in this Standard. Refer to section 0 for supported CSTA events supported by the monitorFilter.
monitorType	Y	Specifies the type of monitor requested. Supported options: • device-type (default if not provided)

Parameter Name	Supported	Comments
requestMonitorMediaClass	N	Specifies the media classes (voice, digital data, Email, message, etc.) of calls that are being requested to be monitored for the monitorObject.

OpenScape Voice Operational Notes

- The monitorObject must be a deviceID.
- OpenScape Voice supports only device monitoring. If the call leaves the monitored device the monitoring can't be continued.
- OpenScape Voice does not currently limit the number of device monitors per TCP link or Application Session.
- Monitoring is only guaranteed for devices within the switching sub domain.
- Multiple monitors may exist for the same station device or group device. Each monitor may have a unique event filter.
- Multiple applications monitoring and controlling the same device are responsible for coordinating any call control functions to avoid conflicts among the applications.
- Group Devices (MLHG) may be monitored and provide backup call distribution (UCD mode) for applications that use Application Session Services.
- Monitoring of ONS provisioned SIP deviceID is supported regardless of whether a physical device is registered for that deviceID.

4.3.1.3 Monitor Stop

The Monitor Stop service is used to cancel a previously initiated Monitor Start service.

The Monitor Stop service can be issued by a function to terminate or signal the termination of a corresponding Monitor Start service.

Service Request Parameters

Parameter Name	Supported	Comments
crossRefIdentifier	Y	This specifies which monitor to cancel. **Note:** In the case of a bulk monitor object "XR:BBBBx" returned in the Monitor Start Response. The "x" is an ordinal number that makes the bulk monitor object unique.

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2.

Service Response Parameters

This service follows the atomic acknowledgement model for this request.

Positive Acknowledgement

A positive acknowledgement to the service request indicates that the Cross Reference ID provided in the Monitor Start response is no longer invalid.

OpenScape Voice Operational Note

The switching function may issue a Monitor Stop service when it can no longer provide information, for example if device is deleted through administration.

4.4 Snapshot Services

4.4.1 Services

4.4.1.1 Snapshot Call

The Snapshot Call service provides information about the devices participating in a specified call. The information provided includes device identifiers, their connections in the call, and local connection states of the devices in the call as well as call related information. Information that applies to the entire call is provided in the Snapshot Call positive response. Information that is specific to each endpoint in the call (snapshotData parameter) is provided in the Snapshot Call positive acknowledgement Dynamic Feature Availability is supported and provided in the servicesPermitted parameter.

Service Request Parameters

Parameter Name	Supported	Comments
snapshotObject	Y	Indicates the connection ID of the call to be snapshot. ECMA also provides the option for CallID only for this parameter; however this option is not supported by OpenScape Voice.

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Positive Acknowledgement

<table>
<tr><th>Parameter Name</th><th>Supported</th><th>Comments</th></tr>
<tr><td>serviceCrossRefID</td><td>Y</td><td>Specifies the reference used to associate subsequent Snapshot CallData services to this service request. OpenScape Voice does not support Snapshot Call Data service.</td></tr>
<tr><td>snapshotData</td><td>Y</td><td>Specifies information for each endpoint in a call. Supported parameters for snapshotDeviceResponseInfo:
<table>
<tr><th>Parameter Name</th><th>Supported</th><th>Type</th></tr>
<tr><td>connectionIdentifier</td><td>Y</td><td>connectionID</td></tr>
<tr><td>deviceID</td><td>Y</td><td>deviceID</td></tr>
<tr><td>localCallState</td><td>Y</td><td>choiceStructure
OpenScape Voice supports compoundCallState
<table>
<tr><th>Parameter Name</th><th>Supported</th><th>Type</th></tr>
<tr><td>compoundCallState</td><td>Y</td><td>localConnectionState</td></tr>
</table></td></tr>
<tr><td>mediaCallCharacteristics</td><td>Y</td><td>Supported values:
• Voice
• Image</td></tr>
<tr><td>servicesPermitted</td><td>Y</td><td>Specifies a list of the call control services that can be applied to the local connection.</td></tr>
<tr><td>ConnectionInformation: flowDirection</td><td></td><td>Supported flowDirection:
• Transmit&Receive
• Receive (silent monitoring active)</td></tr>
</table></td></tr>
<tr><td>callingDevice</td><td>Y</td><td>Specifies the calling device.</td></tr>
<tr><td>calledDevice</td><td>Y</td><td>Specifies the called device</td></tr>
<tr><td>associatedCallingDevice</td><td>Y</td><td>Specifies the Network Interface Device associated with the calling device if the call is an external incoming call. Supported options:
Always “notKnown”
Not provided for outgoing calls.</td></tr>
<tr><td>associatedCalledDevice</td><td>Y</td><td>For outgoing external calls, this parameter specifies the
Network Interface Device associated with the originally called device. For incoming external calls, this parameter specifies a device within the switching sub-domain associated with the originally called device. Not provided for incoming calls.</td></tr>
</table>

OpenScape Voice Operational Notes

- The snapshotObject connectionID must contain a valid call ID and a valid device ID. *ECMA also provides the option for CallID only for this parameter; however this option is not supported by OpenScape Voice.*
- If any of the partner devices are not provisioned with CSTA the servicesPermitted is omitted and the localConnectionInfo becomes “unknown”.
- The servicesPermitted parameter returned in the response may not be fully accurate due to the limitation of information kept internally in OpenScape Voice. Applications should resynchronize their servicesPermitted knowledge on subsequent events is generated for the call.

- Snapshot Call is currently not supported for Network Interface Devices.

4.4.1.2 Snapshot Device

The Snapshot Device service provides information about calls associated with a given device. The information provided identifies each call that the device is participating in and the local connection state of the device in that call. This service also provides a mechanism to cleanup any so-called “ghost calls”.

Service Request Parameters

Parameter Name	Supported	Comments
SnapshotObject	Y	The Device ID is the directory number of the snapshot device.

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Service Response Parameters

<table>
<tr><th>Parameter Name</th><th>Supported</th><th colspan="4">Comments</th></tr>
<tr><td>serviceCrossRefID</td><td>Y</td><td colspan="4">Specifies the reference used to associate subsequent Snapshot CallData services to this service request.</td></tr>
<tr><td rowspan="10">snapshotData</td><td rowspan="10">Y</td><td colspan="4">Specifies information for each endpoint in a call. Supported parameters for snapshotDeviceResponseInfo:</td></tr>
<tr><td>Parameter Name</td><td>Supported</td><td colspan="2">Type</td></tr>
<tr><td>connectionIdentifier</td><td>Y</td><td colspan="2">connectionID</td></tr>
<tr><td>deviceID</td><td>Y</td><td colspan="2">deviceID</td></tr>
<tr><td>localCallState</td><td>Y</td><td colspan="2">choiceStructure
OpenScape Voice supports compoundCallState</td></tr>
<tr><td></td><td></td><td>Parameter Name</td><td>Supported</td><td>Type</td></tr>
<tr><td></td><td></td><td>compoundCallState</td><td>Y</td><td>localConnectionState</td></tr>
<tr><td>mediaCallCharacteristics</td><td>Y</td><td colspan="2">Supported values:
• Voice
• Image</td></tr>
<tr><td>servicesPermitted</td><td>Y</td><td colspan="2">Specifies a list of the call control services that can be applied to the local connection.</td></tr>
<tr><td>privateData</td><td>Y</td><td colspan="2">outOfService; indicates that no SIP devices are registered device. See OutOfService and BackInService events.</td></tr>
</table>

OpenScape Voice Operational Notes

- This service does not affect the call state at the specified device.
- This service reports available for the prime line of a keyset device only.
- If there are <= 32 calls on a device, the response to SnapshotDevice returns the information about all of those calls. If there are > 32 calls on a device, the response to SnapshotDevice returns a cross reference ID and the information about the calls are returned in one or more SnapshotDeviceData messages (32 calls per message).
- localCallState provides simpleCallState view:

  If for example, A consults C with B on hold, the Snapshot calls would return as follows:

  A = 2 calls, hold(1st call), connected(2nd call)
  B = 1 call, connected
  C = 1 call, connected
- Service is only supported for keyset device prime line.
- All active connections known by CSTA are reported immediately. After the acknowledgement, CSTA performs an asynchronous verification on each connection to determine if these connections that are still invalid (i.e., so called ghost-calls). Invalid connections are cleared by CSTA.
- Snapshot Device is currently not supported for Network Interface Devices.
- A QueuedEvent is optionally presented when all members of the group are busy (AMB). If the Hunt Group is in the AMB state then the connectionID is presented in the SnapshotDeviceResponse. Snapshot Call is supported on AMB connection. The following is an example XML of SnapshotCallResponse:

```
<?xml version="1.0" encoding="UTF-8"?>
<SnapshotCallResponse xmlns="http://www.ecma-
international.org/standards/ecma-323/csta/ed4">
 <crossRefIDorSnapshotData>
   <snapshotData>
     <snapshotCallResponseInfo>
       <deviceOnCall>
          <notKnown />
       </deviceOnCall>
       <callIdentifier>
         <callID>FF00020000000000A15D125296080000</callID>
         <deviceID>
           <notKnown />
         </deviceID>
       </callIdentifier>
       <localConnectionInfo>connected</localConnectionInfo>
       <servicesPermitted>
```

```
            <callControlServices />
            <callAssociatedServices />
            <mediaAttachementServices />
            <routeingServices />
            <voiceServices />
          </servicesPermitted>
          <mediaServiceInfoList />
        </snapshotCallResponseInfo>
      </snapshotData>
    </crossRefIDorSnapshotData>
    <callingDevice>
      <notKnown />
    </callingDevice>
    <calledDevice>
      <deviceIdentifier>+15619231416</deviceIdentifier>
    </calledDevice>
</SnapshotCallResponse>
```

4.4.1.3 Snapshot CallData

Service Not Supported

This service is generated as a result of the Snapshot Call service. It is used when the switching function is providing snapshot call information in multiple messages (otherwise the switching function provides the snapshot call information in the Snapshot Call positive acknowledgement).

4.4.1.4 Snapshot DeviceData

This service is generated as a result of the Snapshot Device service. It is used when the switching function is providing snapshot device response information in multiple messages (otherwise the switching function provides the snapshot device response in the Snapshot Device positive acknowledgement).

This includes information about calls associated with a given device. The information provided identifies each call the device is participating in and the local connection state of the device in that call. The switching function may generate a sequence of Snapshot DeviceData services, individually referred to as segments, in response to a single Snapshot Device service request.

Service Request Parameters

Parameter Name	Supported	Comments
serviceCrossRefID	Y	Specifies the reference used to associate the Snapshot DeviceData service messages to the Snapshot Device service request.

<table>
<tr><th>Parameter Name</th><th>Supported</th><th>Comments</th></tr>
<tr><td>segmentID</td><td>Y</td><td rowspan="2">Specifies the segment number of this message. Each successive segment number in the sequence increments the segmentID by one.
Specifies (True / False) if this segment is the last one associated with the serviceCrossRefID.</td></tr>
<tr><td>lastSegment</td><td>Y</td></tr>
<tr><td>snapShotData</td><td>Y</td><td>
<table>
<tr><th>Parameter Name</th><th>Supported</th><th>Type</th></tr>
<tr><td>connection Identifier</td><td>Y</td><td>connectionID</td></tr>
<tr><td>deviceID</td><td>Y</td><td>deviceID</td></tr>
<tr><td>localCallState</td><td>Y</td><td>choiceStructure
OpenScape Voice supports compoundCallState
<table>
<tr><th>Parameter Name</th><th>Supported</th><th>Type</th></tr>
<tr><td>compointCallState</td><td>Y</td><td>localConnectionState</td></tr>
</table>
</td></tr>
<tr><td>mediaCallCharateristics</td><td>Y</td><td>Supported values:
• Voice
• Image</td></tr>
<tr><td>servicesPermitted</td><td>Y</td><td>Specifies a list of the call control services that can be applied to this local connection</td></tr>
</table>
</td></tr>
</table>

Service Response Parameters

There is no service request completion conditions associated with this service.

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

OpenScape Voice Operational Notes

See Snapshot Device notes.

4.5 Application Session Services

Applications are not required to implement Application Session Services to use CSTA services.

Applications Session Services (refer to Section 1.4, “References”) establish and maintain a relationship between an application and a server for the purpose of exchanging application messages. For the purpose of this Standard this relationship is called an application session.

Session IDs provide mechanisms for:

- OpenScape Voice and Application to preserve and recover device monitor cross-reference ID after a link failure.
- Enables Group Device (MLHG) under application control to provide backup call distribution (UCD) based on the application heartbeat extension.
- Enables applications to declare exclusive call control for a Group Device (MLHG).

OpenScape Voice supports the following Application Session Services as defined by ECMA 354. Both HPPC application (A) and OpenScape Voice server (S) must comply with all conformance requirements defined in section 2 of ECMA 354 for:

- Start Application Session (A → S)
- Stop Application Session (A → S)
- Reset Application Session Timer (A → S)
- Application Session Terminated (A ← S)

The following are known limitations in OpenScape Voice implementation of Application Session Services:

- OpenScape Voice supports only one Application Session per CSTA TCP connection. A second Start Application Session Service request on a CSTA TCP connection is rejected.
- System Status messages related to an Application Session are not supported.
- Request System Status messages should not be exchanged for application heartbeat purposes when using an Application Session. Application heartbeat for the purposes of backup call distribution from a Group Device (Hunt) is a FUTURE requirement.

4.5.1 Start Application Session

The Start Application Session service is used to initiate an application session between an application and a server.

A globally unique identifier, called a sessionID, is returned in the positive service response that identifies the application session.

Once an application session is established, the server must maintain information associated with the application session. When the application session is terminated, the application context information is cleared.

The application session exists until:

- It is stopped by using a Stop Application Session service
- The session is abnormally terminated by the server as indicated by the Application Session Terminated service (for example, due to the session timer expiry)

Service Request Parameters

Parameter Name	Supported	Comments
applicationInfo	Y	Specifies information associated with the application requesting the application session. Supported parameters: • applicationID • applicationSpecificInfo
requestedProtocolVersions	Y	Specifies one or more application protocol versions that the application wants to use for the application association. The list is ordered by highest priority protocol version first. • ECMA-323 (Edition 4)
requestedSessionDuration	Y	Specifies the length of time (in seconds) that the application session should be maintained. The sessionDuration timer can be periodically refreshed using the Reset Application Session Timer service. If the requestedSessionDuration is not provided the server chooses a default sessionDuration value. Refer to usage notes for session duration negation rules.
extensions	Y	Specifies non-standardized information. OpenScape Voice supports an application heartbeat timer negotiation which can be used to trigger backup call distribution at a monitored group device (MLHG). Supported options: • requestedHBT value n (see usage notes)

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Negative acknowledgement error values are defined in the service response definition below.

Positive Acknowledgement

Parameter Name	Supported	Comments
sessionID	Y	Specifies the globally unique identifier associated with the application session that has been created. The SessionID size is 9 characters and ensured unique system wide
actualProtocolVersion	Y	Specifies the protocol version that is being used for the application session. This protocol version shall be one of the protocol versions specified in the service request. OpenScape Voice does not currently support protocol negotiation and accepts requestedProtocolVersion as shown below. Any mismatch in namespace and installed release is rejected. • ECMA-323 Edition 4
actualSessionDuration	Y	Specifies the new value for the length of time (in seconds) that the application session is maintained by the server. This value may be less than or equal to the requestedSessionDuration in the service request. If the requestedSessionDuration is not provided in the service request, the server provides a default value which shall be used for the application session. See usage notes for session duration negation rules.
extensions	Y	Specifies non-standardized information. OpenScape Voice supports an application heartbeat timer negotiation which can be used to trigger backup call distribution at a monitored group device (MLHG). Supported options: • actualHBT value n (see usage notes)

Negative Acknowledgement

Parameter Name	Supported	Comments
errorCode	Y	Specifies the type of error. Either a standardized error or an application specific error may be used. The standardized set of errors are: • invalidApplicationInfo – the server is unable to establish an application session due to invalid or unrecognized information in the applicationInfo parameter. • requestedProtocolVersionNotSupported – none of the requested protocol versions specified in the service request are supported. • serverResourcesBusy – the server cannot establish an application session due to internal resource constraints. • maxNumberSessions – the server cannot create an application session because it has reached the maximum number of allowed application sessions (or example, license/provisioning limitations).
extensions	Y	Specifies non-standardized information.

OpenScape Voice Operational Notes

None

4.5.1.1 Basic Application Session (Monitor Recovery)

- The applicationInfo information provided by the application is only logged by OpenScape Voice. No other usage of this information defined for this information.
- Currently Application Session Services have a one-to-one relationship with the application and virtual TCP address provisioned for CSTA service access (Refer to Section 3.1, “Application Connectivity”).
- Application Session XML namespace ECMA 354 was added to all Application Session Services. For the requestedProtocolVersion and actualProtocolVersion, the OpenScape Voice switching function only uses the protocolVersion in the request/response messages exchanged with the application. Therefore, it is required that the applications list the Application Session XML namespace as their first namespace in the protocolVersion element of their Application Session XML messages (refer to Section 2.2.2, “ECMA-323 Specified Tags”).
- The requestedDuration granularity is seconds, if no requestedDuration is provided by the application OpenScape Voice responds to the Start Application Session request with an actualDuration of 900 seconds (15 minutes). This is an OpenScape Voice default value and does not require provisioning.

- OpenScape Voice and application may negotiate for the actualDuration. The requestedDuration and actualDuration granularity is in seconds. OpenScape Voice supports an Application Session duration range of 120 seconds (2 minutes) minimum to 3600 seconds (1 hour) maximum.
- Once the session duration has been successfully negotiated, OpenScape Voice and application must preserve monitor monitorCrossReferenceIDs across temporary link failures or OpenScape Voice restarts for each active sessionID.
- The requestedDuration and actualDuration granularity is in seconds. OpenScape Voice supports an Application Session duration range of 120 seconds (2 minutes) minimum to 3600 seconds (1 hour) maximum.
- The following table defines the OpenScape Voice rules for delivering the actualDuration:

requestedDuration (from application)	**actualDuration (from OpenScape Voice)**	
	Start Application Session	**Reset Application Session Timer**
Within duration range	requestedDuration	requestedDuration
< minimum range value	Minimum duration value	Minimum duration value
> maximum range value	Maximum duration value	Maximum duration value
Not provided by application	900 seconds (15 minutes)	Last negotiated actualDuration

4.5.1.2 Application Session with Application Heartbeat (Fallback Call Distribution)

- The **extensions** parameter defined in ECMA 354 shall for the purpose of negotiating a Heartbeat Timeout value between OpenScape Voice and application.
- OpenScape Voice uses the negotiated Heartbeat value to trigger backup call distribution on specific Group Devices (MLHG).
- The proposed value for extension parameter in the service request is requestedHBT=<value in seconds>. The proposed value for the extension parameter in the service response is actualHBT=<value in seconds>.
- The requestedHBT and actualHBT parameter values shall be independent of the requestedDuration and actualDuration value used for the overall Application Session.
- The requestedHBT is a conditional parameter and should only be used when an application requires backup call distribution timeout on a Group Devices. Only one actualHBT is supported per application session. The actualHBT is

applied to a Group Device when the sessionID is specified by the application in the monitor start for the Group Device. Refer to Section 4.3.1.2, “Monitor Start” for details.

- The rules for used by OpenScape Voice to determine the actualHBT value are as follows:

requestedHBT (from application)	actualHBT (from OpenScape Voice)	
	Start Application Session	**Reset Application Session Timer**
< actualDuration for the session	requestedHBT	requestedHBT
> actualDuration for the session	Use the minimum session duration value	Use the minimum session duration value
Not provided by application	No HBT is used by application	Last negotiated actualHBT

- The application is required to send the Reset Application Session Timer before the HBT expires to maintain control of Group Device and prevent backup call distribution from starting. Both the requestedSessionDuration and HBT are optional. If not supplied the previously used values are to be re-used.
- Expiration of the HBT does not affect the active Application Session which remains active until the actualDuration timer expires.

Note: While ECMA 354 permits multiple application sessions for CSTA over SOAP, this is not the case for CSTA over TCP supported by OpenScape Voice. Only one Application Session per TCP link is supported by OpenScape Voice. OpenScape Voice negatively acknowledges all subsequent Start Application Session service requests received on a TCP link with an existing application session.

OpenScape Voice currently supports a maximum of 16 application links.

- All active Application Sessions are maintained by the OpenScape Voice standby node in the event of a switchover. In case of a switchover, if primary node has active Heartbeat Timers, these are restarted on the secondary node when the secondary node takes over.

4.5.2 Stop Application Session

The Stop Application Session service is used by the application to terminate an existing application session.

Service Request Parameters

Parameter Name	Supported	Comments
sessionID	Y	Specifies the globally unique identifier associated with the application session that is being stopped.
sessionEndReason	Y	Specifies the reason that the application session is being stopped. Either a standardized reason or an application specific reason may be used. The standardized set of reasons are: • normal – the application is no longer interested in maintaining this application association.
extensions	Y	Specifies non-standardized information.

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Negative acknowledgement error values are defined in the service response definition below.

Positive Acknowledgement

Parameter Name	Supported	Comments
extensions	Y	Specifies non-standardized information.

Negative Acknowledgement

Parameter Name	Supported	Comments
errorCode	Y	Specifies the type of error. Either a standardized error or an application specific error may be used. The standardized set of errors are: • invalidSessionID – the sessionID is not valid or known by the server.
extensions	Y	Specifies non-standardized information.

OpenScape Voice Operational Notes

Refer to notes in Section 4.5.1, “Start Application Session”.

4.5.3 Reset Application Session Timer

The Reset Application Session Timer service is used by the application to reset the duration that an existing application session.

Service Request Parameters

Parameter Name	Supported	Comments
sessionID	Y	Specifies the globally unique identifier associated with the application session whose timer is being reset.
requestedSessionDuration	Y	Specifies the new value for the length of time (in seconds) that the application requests that the server maintain the application session.
extensions	Y	Heartbeat timer renegotiation is supported. Refer to usage notes in Section 4.5.1, “Start Application Session”.

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Negative acknowledgement error values are defined in the service response definition below.

Positive Acknowledgement

Parameter Name	Supported	Comments
actualSessionDuration	Y	Specifies the new value for the length of time (in seconds) that the application session is maintained by the server. This value may be less than or equal to the requestedSessionDuration in the service request. If the requestedSessionDuration is not provided in the service request, the server provides a default value which is used for the application session.
extensions	Y	Heartbeat timer renegotiation is supported. Refer to usage notes in Section 4.5.1, “Start Application Session”.

Negative Acknowledgement

Parameter Name	Supported	Comments
errorCode	Y	Specifies the type of error. Either a standardized error or an application specific error may be used. The standardized set of errors are: • invalidSessionID – the sessionID is not valid or known by the server. • serverCannotResetSessionDuration – the server cannot reset the session timer associated with the application session.
extensions	Y	Specifies non-standardized information.

OpenScape Voice Operational Notes

Refer to notes in Section 4.5.1, “Start Application Session”.

4.5.4 Application Session Terminated

The Application Session Terminated service is sent by a server when the server has terminated an application session.

Service Request Parameters

Parameter Name	Supported	Comments
sessionID	Y	Specifies globally unique identifier associated with the application session that is being terminated.
sessionEndReason	Y	Specifies the reason that the application session has been terminated. Either a standardized reason or an application specific reason may be specified. Supported parameters: • resourceLimitation – application session terminated due to resource constraints. • sessionTimerExpired – application session terminated due to the sessionTimer expiry.
extensions	Y	Specifies non-standardized information.

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Response Parameters

None

OpenScape Voice Operational Notes

Refer to notes in Section 4.5.1, “Start Application Session”.

4.6 Call Control

This subsection describes the Call Control features of this Standard. It includes:

- Call Control services
- Call Control events

4.6.1 Services

4.6.1.1 Accept Call (ONS Subscribers Only)

Attention: This section is for Unify internal use only. Refer to Appendix F, “One Number Service (ONS)” for details.

OpenScape Voice supports Accept Call for devices provisioned for ONS-IO service only. An inbound call to an ONS provisioned subscriber remains in the alert-pending state for 2 seconds or until acted upon by the application.

Service Request Parameters

Parameter Name	Supported	Comments	
callToBeAccepted	Y	Specifies the connection to be accepted.	
		Initial State	Supported
		Alerting - Offered	Y
privateData	Y	ONS only. noReOffer is mandatory if Offer Call is “distributed”	

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model. Atomic means that only Offered Timer is stopped and the call progresses to the ONS subscriber’s registering device.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Response Parameters

None

ONS Usage

- Inbound provisioned and active switch services, such as call forwarding, group pickup, etc. will be started.

- If the subscriber device is currently unregistered or unreachable (does not respond to the setup) the call forward-dependable or call forwarding switching function service will be applied after the Accept Call service but before the device is alerted. This results in an immediate Diverted Event.

  Note: It may take up to 30 seconds for the switch to detect that the device is unreachable.

- Call forward-dependable service must be provisioned to address ONS configurations that use non-registering devices exclusively. Call forward-dependable is recommended to address inbound calls to a device that register but may become temporarily unreachable.

4.6.1.2 Alternate Call

The Alternate Call service places an existing active call on hold and then retrieves a previously held call. OpenScape Voice does not currently permit this service usage to place an active call on hold and then connect to an alerting or queued call at the same device (that is, to answer a call-waiting call).

Service Request Parameters

Parameter Name	Supported	Comments	
heldCall	Y	Specifies the held connection for the alternating device.	
		Initial State	Supported
		Alerting - Offered	N
		Alerting - Delivered	Y
		Alerting – Entering Distribution	N
		Hold	Y
		Queued	N
activeCall	Y	Specifies the active connection for the alternating device.	

Acknowledgement Model

OpenScape Voice supports the multi-step positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Response Parameters

None

OpenScape Voice Operational Notes

- Alternate service is only supported for Unify phones. Refer to Section 3.2.2.1, “Type1 and CSTA over SIP Device Type configuration”.
- This service is supported for ONS-IO.

Attention: One Number Service (ONS) is internal to Unify and is used over the CSTA interface between OpenScape Voice and OpenScape UC Application (Refer to Appendix F, “One Number Service (ONS)” for details.

4.6.1.3 Answer Call

The Answer Call service connects an alerting call. This service is typically associated with devices that have attached speakerphone units and headset telephones to connect to a call using hands-free operation.

Service Request Parameters

<table>
<tr><th>Parameter Name</th><th>Supported</th><th colspan="2">Comments</th></tr>
<tr><td rowspan="6">callToBeAnswered</td><td rowspan="6">Y</td><td colspan="2">Specifies the connection to be answered.</td></tr>
<tr><td>Initial State</td><td>Supported</td></tr>
<tr><td>Alerting - Offered</td><td>N</td></tr>
<tr><td>Alerting - Delivered</td><td>Y</td></tr>
<tr><td>Alerting – Entering Distribution</td><td>N</td></tr>
<tr><td>Initiated</td><td>N</td></tr>
<tr><td>Queued</td><td>N</td></tr>
</table>

Acknowledgement Model

OpenScape Voice supports the multi-step positive acknowledgement model. Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Response Parameters

None

OpenScape Voice Operational Notes

- Answer Call is possible if the called device has only 1 primary line registration active and no other secondary or phantom lines registered.

Note: The called primary line DN may appear as secondary lines on other devices.

- If more than 1 contact is registered on the alerting device DN then Answer call is not possible.
- CSTA supports the following first party device control options for Answer Call in the following order depending on device support:

  a) uaCSTA

  b) Remote Control Talk Event Package

  c) Alert-Info header-field with Auto-Answer indication. Not supported for callback recall.
- If a call waiting call is answered the currently connected call is placed on hold and the alerting call waiting call is answered.
- This service is supported for ONS.

4.6.1.4 Call Back Call-Related

The Call Back Message Call-Related service allows a computing function to request that the switching function leave a pre-defined message requesting that the called device call the calling device. For example, the called device may have been busy when called.

Service Request Parameters

Note: This service is only allowed for alerting or busy calls.

<table>
<tr><th>Parameter Name</th><th>Supported</th><th colspan="2">Comments</th></tr>
<tr><td rowspan="11">callbackConnection</td><td rowspan="11">Y</td><td colspan="2">Specifies the callback connection at the calling device.
The callback connection may include ONS binding (refer to operational note 3.)</td></tr>
<tr><td>Initial Local Connection Information</td><td>D1C1</td></tr>
<tr><td>Alerting - Offered</td><td>N</td></tr>
<tr><td>Alerting - Delivered</td><td>Y</td></tr>
<tr><td>Alerting - Entering Distribution (Hunt Group)</td><td>N</td></tr>
<tr><td>Connected</td><td>*N1</td></tr>
<tr><td>Failed</td><td>Y</td></tr>
<tr><td>Hold</td><td>N</td></tr>
<tr><td>Initiated</td><td>N</td></tr>
<tr><td>Queue</td><td>N</td></tr>
<tr><td colspan="2">* if D2 is across a network interface that supports callback service then callback is possible in connected state on N1C1. OpenScape Voice represents N1 as D1.</td></tr>
<tr><td>callCharacteristics</td><td>N</td><td colspan="2">Specifies the high level characteristics</td></tr>
<tr><td>SubjectOfCall</td><td>N</td><td colspan="2">Specifies the subject/intent of the call</td></tr>
<tr><td>languagePreference</td><td>N</td><td colspan="2">Specifies the language preferences to be associated with the call</td></tr>
<tr><td>security</td><td>N</td><td colspan="2">Specifies timestamp information, message number, and security information</td></tr>
<tr><td>privateData</td><td>N</td><td colspan="2">Specifies non-standardized information</td></tr>
</table>

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model.

Positive Response Parameters

Parameter Name	Supported	Comments
targetDevice	Y	Specifies the deviceID of the device that the call back was initiated for (See operational note 4).
security	N	Specifies timestamp information, message number, and security information.
privateData	N	Specifies non-standardized information.

Response Parameters

If Callback Call-Related service cannot be invoked OpenScape Voice V9 shall respond with an appropriate CSTA Error.

Possible CSTA errorValues include:

- invalidConnectionState – if callbackConnection is not alerting or busy state
- requestIncompatibleWithDevice – targetDevice (alerting or busy device) does not support callback service
- invalidCallID – unknown callID
- privilegeViolationSpecifiedDevice: There is a limit in callback requests that a user can initiate. If a user exceeds this limit the response will be this one; Or the callback cannot be activated temporarily. In a short period of time a reactivation may be successful. Or the callback cannot be activated for a long period of time.
- featureAlreadySet: A callBack request already exists on a target device
- invalidDeviceID: CCS fails to verify the FQN from the mem slot vs the FQN from the target device.

OpenScape Voice Operational Notes

- The calling device’s local connection state must be connected.
- The target device's local connection state must be Delivered (alerting) or Failed (e.g., Busy).

Note: D2 is across a network interface that supports callback service then callback is possible in connected state on N2C1.

- The targetDevice in the service response shall be presented in FQN (if possible based on provisioning of number translation).
- If callback connection includes media other than voice (e.g., image) the call back shall be set but only voice media shall be invoked in the callback-recall setup.

4.6.1.5 Camp on Call

Service Not Supported

The CampOn Call service allows the computing function to queue a call for a device (that typically is busy) until that device becomes available (after finishing a current call or any previously queued calls, for example).

4.6.1.6 Clear Call

Service Not Supported

The Clear Call service releases all devices from an existing call. In the case of a conference call, this results in all devices in the conference call being released from the call.

4.6.1.7 Clear Connection

The Clear Connection service releases a specific device from a call. In the case of a two-party call, this may result in the call being torn down. In the case of a conference call, this results in the specific party being removed from the conference.

Clear Connection can also be used on a connection in the Offered state to effectively reject the call.

Service Request Parameters

Parameter Name	Supported	Comments		
connectionToBeCleared	Y	Specifies the connection to be cleared.		
		Initial State	**Supported**	**Note**
		Alerting - Offered	Y	10
		Alerting - Delivered	Y	4
		Alerting – Entering Distribution	Y	
		Connected	Y	
		Failed	Y	
		Hold	Y	1, 7
		Initiated	Y	
		Queued	Y	7
reason	N	Specifies the reason the connection is being cleared (busy, for example).		

Acknowledgement Model

OpenScape Voice supports the multi-step positive acknowledgement model.

Negative acknowledgement error values are defined inSection 2.6.2, “Negative Service Response”.

Response Parameters

None

Miscellaneous Characteristics

DeviceIDOnly -The switching function accepts and supports the DeviceID only format of the Connection ID for this service.

OpenScape Voice Operational Notes

1. The Clear Connection service does not automatically retrieve a held call. If an application wants to clear an active call and retrieve a currently held call, the Reconnect service must be used instead of Clear Connection.
2. The Clear Connection service results in a negative acknowledgement if the clearing party is connected to a conference and has a call on soft hold.

3. If the clearing device's Device ID refers to an extension that appears on more than one device (multiple appearance), the Clear Connection service is attempted only on the device where the specified extension number is configured as its primary device, and only if no other appearances are active (that is, the Connection State is *connected*) on that device. If these conditions are not met, a negative response is returned. In addition, if an appearance of the primary device is connected into the call (bridged conference), a negative response is returned.
4. If the primary device is ringing, other appearances of that line are also cleared.
5. The clearing party cannot be the target party of an executive override.
6. After the successful execution of the Clear Connection service, the clearing device and any non-conferenced devices in the call transition from connected, to failed, and to null or connected to null.
7. A held or queued connection can only be cleared if there is no other active connection at the clearing device.
8. The Clear Connection service results in a negative acknowledgement if the clearing party has more than one connection and the service request was initiated with DeviceIDOnly.
9. If the target device is a SIP phone that supports uaCSTA, OpenScape Voice may use uaCSTA to clear the connection at this device. The CSTA application has no control if uaCSTA is used or not.
10. Clear Connection does not use uaCSTA if the call is in the Offered state. At this point the call is not established at a physical device.
11. Clear Connection invoked on a Callback Recall cancels the original callback.

4.6.1.8 Conference Call

The Conference Call service provides a conference of an existing held call and another active call at a conferencing device.

The two calls are merged into a single call and the two connections at the conferencing device are resolved into a single connection. The Connection IDs formerly associated with the conferenced connections are released and a new Connection ID for the resulting connection is created. The existing held call may consist of two or more devices.

Service Request Parameters

Parameter Name	Supported	Comments	
heldCall	Y	Specifies the held connection.I	
		Initial State	Supported
		Alerting - Offered	N
		Alerting - Delivered	N
		Alerting – Entering Distribution	N
		Connected	N
		Hold	Y
		Queued	N
activeCall	Y	Specifies the active connection.	
		Initial State	Supported
		Connected	Y
		Hold	N

Acknowledgement Model

OpenScape Voice supports the multi-step positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Response Parameters

Parameter Name	Supported	Comments
conferenceCall	Y	Specifies the resulting connection to the new call. The ConnectionID shall have the CallID of the resulting conference call and the DeviceID of the conferencing device.

OpenScape Voice Operational Notes

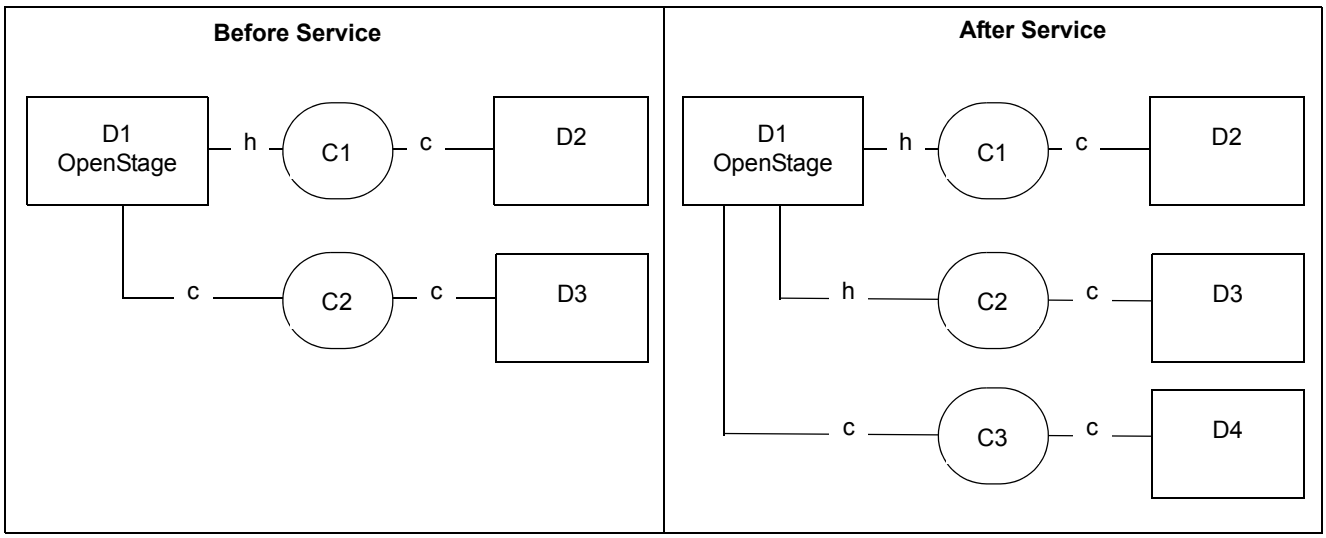
1. Adding a party to an *existing* conference is only supported for Unify phones. Refer to Section 3.2.2.1, “Type1 and CSTA over SIP Device Type configuration”.
2. CSTA Subscriber must be provisioned for Large Conference Service.

3. Large Conference service requests may be rejected based on limitations and constraints such as:
   - Subscriber is not provisioned for large conference service
   - Three-way call may not be supported when active leg is the result of incoming call, for example. call waiting or pickup group.
   - Number of participants would be exceeded. SIP Large Conference Service supports from 3 to 48 conference members.
   - Media Service pool resources not available
4. Only voice media is supported by the Large Conference Service.
5. Service is not supported for control of SIP local device conferencing.
6. Service is not support to invoke a bridged appearance.

4.6.1.9 Consultation Call

This service places an existing active call at a device on hold and initiates a new call from the same device. The existing active call may include two or more devices (e.g., conference).

OpenStage phones support the capability to invoke a 3rd call leg after answering a waiting call or picking a call during an existing call. A 3rd call leg may be invoked via the Consultation Call service. Refer to PART II for detailed event flows.



Service Request Parameters

Parameter Name	Supported	Comments	
existingCall	Y	Specifies the active connection.	
		Initial State	**Supported**
		Connected	Y
		Hold	Y

Parameter Name	Supported	Comments

consultedDevice	Y	• Specifies the device to be consulted. The number must be dialable by the callingDevice, i.e. no toll or subscriber restriction
		• The number may be a feature access code if calling subscriber is authorized to use the feature. If feature requires subsequent digits to complete the request they must be include in this parameter.
		• It is recommended that features supported via CSTA services, for example, Conference, Silent Monitoring (Join), Callback, Call Completion services, Agent State (Hunt Make Busy), Call Forwarding, Caller Id features, Pickup, be activated by the specific CSTA service.
		• Feature access codes supported:
		  • System Speed Dialing First List
		  • System Speed Dialing Second List
		  • Activate Class Of Service Switchover
		  • Deactivate Class Of Service Switchover
		  • Call Forwarding Override
		  • Caller Identity Suppression Per Call
		  • Activate Calling Identity Suppression
		  • Deactivate Calling Identity Suppression
		  • Toggle Calling Identity Suppression
		  • Alternative Calling Identity
		  • Activate Anonymous Caller Rejection
		  • Deactivate Anonymous Caller Rejection
		  • Malicious Call Trace
		  • Last Incoming Number Redial
		  • Last Outgoing Number Redial
		  • 1-Way Intercom Call (Speaker Call)
		  • 2-Way Intercom Call (Speaker Call)
		  • 1-Way Broadcast Call (Speaker Call)
		  • Business Group Night Bell Call Pickup
		  • Call Park to Server
		  • Call Park Retrieve from Server
		  • Silent Monitoring - Toggle Whisper
		  • Account Code Entry
		  • Pre-Dial Authorization Code
		  • Executive Busy Override
		  • Code Executive Busy Override

Parameter Name	Supported	Comments
userData	Y	Specifies the user data to be sent to parties in a call. Notes: 1. OpenScape Voice V9 currently supports user data invocation for the purpose of sending an application provided “GUID” in the outgoing SIP setup to the consulted device. 2. OpenScape Voice V9 does NOT propagate the application provided “GUID” in subsequent event flow. 3. The “GUID” shall be inserted in the string element and it shall be encoded in hexadecimal value. The maximum length supported by OpenScape Voice V9 is 256 characters (provided via the capabilities exchange services).
accountCode	N	Specifies the account code to associate with the consulted call.
authCode	N	Specifies the authorization code to allow the call.
callCharacteristics	N	Specifies the high level characteristics (Priority call, for example) to be associated with the call.
mediaCallCharacteristics	Y	This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. Supported values: • Voice • Image **Note:** Voice and Image are supported to create a new video call for this service.
subjectOfCall	N	Specifies the subject/intent of the call.
privateData	Y	Application Provided Calling Name and Number privateData tags” • appCallbackName • appCallbackID

Acknowledgement Model

OpenScape Voice supports the multi-step positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Response Parameters

Parameter Name	Supported	Comments
initiatedCall	Y	Specifies the initial connection to the new call. The ConnectionID shall have the CallID of the resulting new call and the DeviceID of the consulting device.

Miscellaneous Characteristics

Multistage dialing is ***not*** supported.

OpenScape Voice Operational Notes

1. Prior to the consultation, the connection at the consulting device must be connected or a negative response is returned.
2. After the successful execution of the consultation, the active connection at the consulting device is in the connected state, and the consultation-held connection at the consulting device is in the hold state (HELD is shown in the phone).
3. After successful execution of the consultation, the connection at the consulted device are alerting, queued (if consulted device is a hunting group), or null (if a feature is active such as call forwarding immediate).
4. If the active call has been placed on consultation hold, but OpenScape Voice is unable to extend the call to the called device, the application must use the Reconnect service to retrieve the party on consultation hold.
5. If the consulting device's Device ID refers to an extension number that appears on more than one device (multiple appearance), the Consultation Call service is attempted only on the device where the specified extension number is configured as its primary device, and only if no other appearances are active (that is, the Connection State is *connected*) on that device.
6. The SIP phone of the holding party is in HELD state, and all restrictions of phone features imposed by this phone state are still applicable.
7. The audible tone or announcements depend on provisioning and if tones are provided by endpoint or OpenScape Voice. If held leg is controlled by OpenScape Voice Conference service no tones or announcements are provided to the held conferees.
8. If the physical consulting device supports uaCSTA (e.g., OpenStage) then it may be possible to invoke a 3rd call leg for the active call via the Consultation Call service.

Note: Consultation call service will be presented in the servicesPermitted parameter for active call when the phone supports uaCSTA. However, the phone will accept a uaCSTA Consultation Call request only under the following conditions:
- after answering a waiting call
- after a Group or Direct Pickup during an active call

This is a known limitation which may cause the Consultation Call request to be rejected.

4.6.1.10 Deflect Call

The Deflect Call service allows the computing function to divert a call to another destination that may be inside or outside the switching sub-domain.

Service Request Parameters

Parameter Name	Supported	Comments	
callToBeDiverted	Y	Specifies the connection to be diverted.	
		Failed	**Supported**
		Alerting - Offered	Y
		Alerting - Delivered	Y
		Alerting – Entering Distribution	Y
		Connected	N
		Failed	N
		Hold	N
		Queued	Y
newDestination	Y	Specifies the device to which the call is to be diverted.	
accountCode	N	Specifies the account code to associate with the consulted call.	
authCode	N	Specifies the authorization code to allow the call.	
callCharacteristics	N	Specifies the high level characteristics (Priority call, for example) to be associated with the call.	
mediaCallCharacteristics	N	This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. This parameter is ignored by OpenScape Voice on the service request and the existing media is retained diverted call.	
subjectOfCall	N	Specifies the subject/intent of the call.	
privateData	Y	ONS: noReOffer element is mandatory if Offered Call cause = distributed.	

Acknowledgement Model

OpenScape Voice supports the multi-step positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Response Parameters

None

OpenScape Voice Operational Notes

1. This service is supported for ONS-IO.

   Attention: One Number Service (OND) is internal to Unify. Refer to Appendix F, “One Number Service (ONS)” for details.

2. Currently CSTA Deflections to internal or on-net devices do not provide loop protection.

4.6.1.11 Dial Digits

Service Not Supported

The Dial Digits service allows the computing function to perform a dialing sequence that is associated with a call that has already been initiated (that is, has manually gone off-hook or has been initiated using a Make Call or Consultation Call service). This service is also used to perform the dialing sequences associated with completing a multi-stage dialed call.

Note: See Generate Digits in Section 4.7.1.1, “Associate Data”.

4.6.1.12 Directed Pickup Call

Service is Not supported

The Directed Pickup Call service moves a specified call and connects it at a new specified destination.

Refer to PART II of the CSTA Interface Specification for manual phone activation of Call Pickup - Directed CSTA event flow.

4.6.1.13 Group Pickup Call

Service is Not supported

The Group Pickup Call service moves a call that is a member of a specified or default pickup group to a new specified destination. This results in a connection in a pickup group to be connected to a new specified destination inside the switching sub-domain.

Refer to PART II of the CSTA Interface Specification for manual phone and SIP-Q (Network) activation of Call Pickup - Group event flow.

4.6.1.14 Hold Call

The Hold Call service places a connected connection on hold at the same device.

Service Request Parameters

Parameter Name	Supported	Comments	
callToBeHeld	Y	Specifies the active connection to be held.	
		Initial State	**Supported**
		Connected	Y

Acknowledgement Model

OpenScape Voice supports the multi-step positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Response Parameters

None

OpenScape Voice Operational Notes

- This service interrupts communication for an existing call at a device. The relationship between the holding device and the held call is maintained until the call is retrieved from the held state or until the call is cleared.
- This service currently does not provide a CSTA recall time-out. Recall is handled by phone or OpenScape Voice service directly.
- Only one call can be put on hold using the Hold Call service. While this call remains on hold, subsequent Hold Call service requests are denied with a negative response.
- This call must be in the connected state on the devices’ primary line or a negative response is returned. In a consultation call scenario, the holding device may be the consulted party. In a conference call scenario, the holding device must be a conference member or the party who was consulted by a conference member.
- If the target device is a SIP phone that supports uaCSTA, OpenScape Voice may use uaCSTA to place the call on hold. If uaCSTA request fails, OpenScape Voice V9 uses 3PCC option to invoke hole.
- The audible tone or announcements depend on provisioning and if tones are provided by endpoint or OpenScape Voice. If held leg is controlled by OpenScape Voice Conference service no tones or announcements are provided to the held conferees.

- This service is supported for ONS-IO.

Attention: One Number Service (ONS) is internal to Unify. Refer to Appendix F, “One Number Service (ONS)” for details.

4.6.1.15 Intrude Call

Service Not Supported

The Intrude Call service adds the calling device to a call at a busy called device. Depending upon the switching function, the result is that the calling device is either actively or silently participating in the called device’s existing call or consulting with the called device with a new call.

4.6.1.16 Join Call

The Join Call service allows a computing function to request, on behalf of a device, that the device be joined into an existing call. In the process of establishing a connection with the joining Device, the joining Device may be prompted to go off-hook (if necessary) and when that device does so, it is added into the call. This service is different from the Single Step Conference service in that the request is made on behalf of the joining Device (originating device). This service is different from the Intrude Call service in that there is no prior failed call at the intruded upon connection.

Service Request Parameters

<table>
<tr><th>Parameter Name</th><th>Supported</th><th colspan="2">Comments</th></tr>
<tr><td rowspan="9">activeCall</td><td rowspan="9">Y</td><td colspan="2">Specifies an existing connection in an active call to which the new device is to be added (or joined).</td></tr>
<tr><th>Initial State</th><th>Supported</th></tr>
<tr><td>Alerting - Offered</td><td>N</td></tr>
<tr><td>Alerting - Delivered</td><td>N</td></tr>
<tr><td>Alerting - Entering Distribution</td><td>N</td></tr>
<tr><td>Connected</td><td>Y</td></tr>
<tr><td>Failed</td><td>N</td></tr>
<tr><td>Hold</td><td>N</td></tr>
<tr><td>Queue</td><td>N</td></tr>
</table>

Parameter Name	Supported	Comments
joiningDevice	Y	Specifies the device that is to be added to (join) the existing call. **Note:** OpenScape Voice V9 supports device-only option for this parameter. This is used to invoke the Continuous Silent Monitoring variant.
autoOriginate	Y	Specifies if the joining device is to be prompted or not (hands-free mode). • Prompt • Do Not Prompt (auto originate) - Default for Join Call service
participationType	Y	Specifies the type of participation the joining device has in the resulting call. • **Silent** - the joining device can listen but cannot actively participate in the resulting conference call. As a result, the flow direction of the joiningDevice's connection (i.e., conferenced call) will be Receive. In this configuration, when either the calling or the called party hangs up or Clear the Connection via CSTA, the conference bridge is released automatically. • **Active** (Default) - the joining device can actively participate in the resulting conference call. As a result, the flow direction of the joiningDevice's connection (i.e., conferenced call) will be Transmit & Receive. When a device is joined actively, each connection to the conference is independent, i.e. the conference bridge shall not be dropped until all parties hang up or a ClearConnection is received on each individual connection.

Acknowledgement Model

OpenScape Voice supports the multi-step positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Response Parameters

Parameter Name	Supported	Comments
conferenceCall	Y	Specifies the callID of the existing active call and the DeviceID of the joining device.

Parameter Name	Supported	Comments
conferenceCallInfo	Y	Specifies the connection information associated with the conferenced call connection, i.e. for silent monitor and whisper Supported values for mediaSessionInfo • Voice • Image Supported values for flowDirection • Transmit - e.g. one-way speaker call • Receive - e.g. silent monitor service • Transmit&Receive - e.g. barge-in; whisper • None **Note:** numberOfChanges is not supported

Join Call Event Flow (Basic)

Refer to CSTA Call Scenarios for additional detailed flows.

OpenScape Voice Operational Notes

- Silent Monitoring and Recording services are supported by manual dialing access code. Refer to CSTA Call Scenarios for event flows.
- Monitored connection may be transferred.
- It is possible to join an active conference.
- It is possible to change the flow direction via the Change Connection Information call associated service. For example, you can change the flow direction on the silent monitoring device from “Receive” to “Transmit&Receive”.
- Continuous Silent Monitor is supported by OpenScape Voice V9. To active Continuous Silent Monitor, the application provides only the device portion of the activeCall connectionID.
- This service is supported for ONS-IO.

Attention: One Number Service (ONS) is internal to Unify. Refer to Appendix F, “One Number Service (ONS)” for details.

4.6.1.17 Make Call

The Make Call service allows the computing function to set up a call between a calling device and a called device.

The service creates a new call and establishes an initiated or connected connection with the calling device. The Make Call service assigns a ConnectionID to the calling device and returns it in the positive acknowledgement.

Service Request Parameters

<table>
<tr><th>Parameter Name</th><th>Supported</th><th colspan="2">Comments</th></tr>
<tr><td rowspan="4">callingDevice</td><td rowspan="4">Y</td><td colspan="2">Specifies the calling/originating device.
OpenScape Voice supports only station deviceID in make call request. This deviceID must be the service ID for the provisioned subscriber DN.</td></tr>
<tr><td>Initial State</td><td>Supported</td></tr>
<tr><td>Null</td><td>Y</td></tr>
<tr><td>Initiated</td><td>N</td></tr>
</table>

Parameter Name	Supported	Comments

calledDirectoryNumber	Y	Specifies the called device.
		• The number must be dialable by the callingDevice, i.e. no toll or subscriber restriction
		• The number may be a feature access code if calling subscriber is authorized to use the feature. If feature requires subsequent digits to complete the request they must be include in this parameter.
		• It is recommended that features supported via CSTA services, for example, Conference, Silent Monitoring (Join), Callback, Call Completion services, Agent State (Hunt Make Busy), Call Forwarding, Caller Id features, Pickup. be activated by the specific CSTA service.
		• Feature access codes supported:
		  • System Speed Dialing First List
		  • System Speed Dialing Second List
		  • Activate Class Of Service Switchover
		  • Deactivate Class Of Service Switchover
		  • Call Forwarding Override
		  • Caller Identity Suppression Per Call
		  • Activate Calling Identity Suppression
		  • Deactivate Calling Identity Suppression
		  • Toggle Calling Identity Suppression
		  • Alternative Calling Identity
		  • Activate Anonymous Caller Rejection
		  • Deactivate Anonymous Caller Rejection
		  • Malicious Call Trace
		  • Last Incoming Number Redial
		  • Last Outgoing Number Redial
		  • 1-Way Intercom Call (Speaker Call)
		  • 2-Way Intercom Call (Speaker Call)
		  • 1-Way Broadcast Call (Speaker Call)
		  • Business Group Night Bell Call Pickup
		  • Call Park to Server
		  • Call Park Retrieve from Server
		  • Silent Monitoring - Toggle Whisper
		  • Account Code Entry
		  • Pre-Dial Authorization Code
		  • Executive Busy Override

accountCode	N	Specifies the account code to associate with the consulted call.
authCode	N	Specifies the authorization code to allow the call.
autoOriginate	Y	Specifies if the calling device's connection is automatically answered (hands-free mode). The complete set of possible values is: • Prompt • Do Not Prompt (auto originate) – Default for Make Call service using OpenScape Voice 3PCC with a Unify phone
callCharacteristics	N	Specifies the high level characteristics (Priority call, for example) to be associated with the call.

Parameter Name	Supported	Comments
mediaCallCharacteristics	Y	This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. Supported values: • Voice • Image **Note:** Voice and Image are supported to create a new video call for this service.
subjectOfCall	N	Specifies the subject/intent of the call.
reason	N	Specifies the reason the connection is being cleared (busy, for example).
privateData	Y	Non-standardized information. • Application Provided Calling Name and Number privateData tags: appCallbackName and appCallbackID, respectively • postDial: enables the provided digits to be dialed after the calledDirectoryNumber device is connected. • The uaCSTA MakeCall Request shall be extended to include the “Use3PCConError” private data extension to indicate fallback to 3PCC MakeCall. **Note:** Note: OpenStage device and OpenScape Voice V9 shall be enhanced to prompt the originating user before establishing a CSTA MakeCall based on the auto-answer configuration on the OpenStage device. If auto-answer is disabled, the phone will alert and the user must manually answer the call before the call is allowed to proceed to the b-leg. The requirement to support 3PCC MakeCall even though the device supports uaCSTA MakeCall will result in increased media server usage to play ringback during call establishment. Solution team must plan for additional media server channels depending on the following factors, BHCA per subscriber, total number of subscribers. OpenScape Voice V9 Capacity planning tool should be used by service and planning team to determine the actual media server requirements for such a deployment with support from Systems Engineering.

Acknowledgement Model

OpenScape Voice V9 supports the multi-step positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

The following error calling device errors are provided:

- Calling device is not registered:
  - CallingDevice is A (may have ONS or not) and A is unregistered: CSTAErrorCode==callingDeviceOutOfService
  - CallingDevice is A|A’ and A’ is unregistered: CSTAErrorCode== callingDeviceOutOfService
- Calling device is not an OpenScape Voice V9 subscriber:
  - CallingDevice is A, A is not an OpenScape Voice V9 subscriber: CSTAErrorCode== invalidCallingDeviceID
  - CallingDevice is A|A’, and translation on DN A’ fails: CSTAErrorCode== invalidDeviceId
- Calling device is an OpenScape Voice V9 subscriber but not CSTA Feature:
  - CallingDevice is A, A is an OSV subscriber but no CSTA feature: CSTAErrorCode==privilegeViolationSpecifiedDevice
  - CallingDevice is A|A’, and A is an OpenScape Voice V9 subscriber with CSTA, but without ONS, CSTAErrorCode==requestIncompatibleWithCallingDevice
- Calling device (or OND) is not reachable:
  - CallingDevice is A, but A is not reachable: Failed even cause== networkNotObtainable
  - CallingDevice is A|A’, A’ is not reachable: Failed event cause==networkNotObtainable
- Calling device (or OND) is busy:
  - CallingDevice is A, but A is busy: Failed Event cause == busy
  - CallingDevice is A|A’, but A’ is busy: Failed event cause== busy

Response Parameters

Parameter Name	Supported	Comments
callingDevice	Y	Specifies the initial connection to the new call. The ConnectionID shall have the CallID of the resulting new call and the DeviceID of the calling device. Note that the calling device parameter in the service request may be different from the deviceID in the connectionID of callingDevice in the positive acknowledgement (when the callingDevice in the service request represents a group of stations, for example).

Miscellaneous Characteristics

- Auto Originate default may be overridden by the application, prompting for calling device is supported.
- MakeCall while calling device is off-hook is supported but may be blocked by SIP device configuration (i.e. busy while dialing).
- MakeCall for a group device is not supported (for example, Hunt Group – MLHG)

OpenScape Voice Operational Notes

- Auto originate = doNotPrompt is possible if the calling device has only 1 primary line registration active and no other secondary or phantom lines registered.

  Note: The primary line DN may appear as secondary lines on other devices.

- The uaCSTA Make Call service may be used to originate registered calling device supports uaCSTA, e.g., Unify OpenStage phone.
- This service is supported for ONS-IO.

  Attention: One Number Service (ONS) is internal to Unify. Refer to Appendix F, “One Number Service (ONS)” for details.

4.6.1.18 Make Predictive Call

Service Not Supported

The Make Predictive Call service shall originate a call between two devices by first creating a connection to the called device. The service returns a positive acknowledgement that provides the connection at the called device.

4.6.1.19 Park Call

Service Not Supported

The Park Call service moves a specified call at a device to a specified (parked-to) destination. The device on whose behalf Park Call is invoked (the parking device) is no longer associated with the call (except when the parking device parks a call back to the parking device).

Note: OpenScape Voice supports a manual Park to Server feature. The Park-to-server feature is activated by creating a consultation call to Park-to-server access code. The Park-to-server scenarios expose a **hard-coded** device ID in the CSTA event flow to represent the system park device. The specific parking space is presented in the displayNumber tag in the NM field. For example, if the space number is #10, then the deviceID would be presented as: ***N<ABC1>Park Slot;displayNumber=10***. For details refer to Park-to-Server event flows in PART II of the CSTA Interface Specification (Refer to Appendix A, “CSTA Call Scenarios”)

4.6.1.20 Reconnect Call

The Reconnect Call service clears a specified connection at the reconnecting device and retrieve a specified held connection at the same device.

Service Request Parameters

Parameter Name	Supported	Comments	
activeCall	Y	Specifies the connection to be cleared.	
		Initial State	Supported
		Alerting - Offered	N
		Alerting - Delivered	N
		Alerting – Entering Distribution	N
		Connected	Y
		Failed	N
		Queued	N
heldCall	Y	Specifies the held connection.	
		Initial State	Supported
		Hold	Y

Acknowledgement Model

OpenScape Voice supports the multi-step positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Response Parameters

None

OpenScape Voice Operational Notes

- Reconnect service is only supported for Unify phones. Refer to Section 3.2.2.1, “Type1 and CSTA over SIP Device Type configuration”.
- If the reconnecting device’s Device ID refers to an extension number that appears on more than one device (multiple appearances), the Reconnect Call service is attempted only on the device where the specified extension number is configured as its primary device, and only if no other appearances are active (that is, the Connection State is *connected*) on that device.
- This service is supported for ONS-IO.

Attention: One Number Service (ONS) is internal to Unify. Refer to Appendix F, “One Number Service (ONS)” for details.

4.6.1.21 Retrieve Call

The Retrieve Call service connects a specified held connection.

Service Request Parameters

Parameter Name	Supported	Comments	
callToBeRetrieved	Y	Specifies the held connection to be retrieved.	
		Initial State	**Supported**
		Hold	Y

Acknowledgement Model

OpenScape Voice supports the multi-step positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Response Parameters

None

OpenScape Voice Operational Notes

- In the case of a manual hold invoked by the endpoint, this service requires that the phone type be Type1 or uaCSTA over SIP Section 3.2.2.1, “Type1 and CSTA over SIP Device Type configuration”].
- If the retrieving device’s DeviceID refers to an extension number that appears on more than one device (multiple appearance), the Retrieve Call service is only attempted on the device where the specified extension number is configured as its primary device, and only if no other appearances are active on that device.
- If the target device is a SIP phone that supports uaCSTA, OpenScape Voice may use uaCSTA to retrieve this call. The CSTA application has no control if uaCSTA is used or not.
- This service is supported for ONS-IO.

Attention: One Number Service (ONS) is internal to Unify. Refer to Appendix F, “One Number Service (ONS)” for details.

4.6.1.22 Send Message

Service Not Supported

The Send Message service allows the computing function to send a message to one or more devices. The message, composed of one or more MIME body parts, is included in the Send Message service request.

4.6.1.23 Single Step Conference Call

Service Not Supported

The Single Step Conference Call joins a new device into an existing call.

This service can be repeated to make n-device conference calls (subject to switching function limits). This service is distinguished from the Join Call service by the way the device being added to the call perceives the direction of the resulting connection (for example, alerts). In the case of the Join Call service, the device generates an outgoing connection (for example, prompts). This affects the parameters associated with the service and the events flowing as a result of the service.

4.6.1.24 Single Step Transfer Call

The Single Step Transfer Call service transfers an existing connected connection at a device to another device.

This transfer is performed in a single-step, where the device doing the transfer does not have to place the existing call on hold before issuing the Single Step Transfer Call service.

Service Request Parameters

<table>
<tr><th>Parameter Name</th><th>Supported</th><th>Comments</th></tr>
<tr><td rowspan="4">activeCall</td><td rowspan="4">Y</td><td colspan="2">Specifies the connected connection in the call to be replaced</td></tr>
<tr><td>Initial State</td><td>Supported</td></tr>
<tr><td>Connected</td><td>Y</td></tr>
<tr><td>Held</td><td>Y</td></tr>
<tr><td>transferredTo</td><td>Y</td><td colspan="2">Specifies the new called (transferredTo) device.</td></tr>
<tr><td>userData</td><td>Y</td><td colspan="2">Specifies the user data to be sent to parties in a call.

Notes:
1. OpenScape Voice V9 currently supports user data invocation for the purpose of sending an application provided “GUID” in the outgoing SIP setup to the transferredTo device.
2. OpenScape Voice V9 does NOT propagate the application provided “GUID” in subsequent event flow.
3. The “GUID” shall be inserted in the string element and it shall be encoded in hexadecimal value. The maximum length supported by OpenScape Voice V9 is 256 characters (provided via the capabilities exchange services).</td></tr>
<tr><td>accountCode</td><td>N</td><td colspan="2">Specifies the account code to associate with the consulted call.</td></tr>
<tr><td>privateData</td><td>Y</td><td colspan="2">Non-standardized information.
• transferTarget values supported:
– voicemail – transfers call directly to the voicemail box of the transferredTo device.
– auto-answer – used by OSMO to invoke auto-answer on the transferredTo device</td></tr>
<tr><td>reason</td><td>N</td><td colspan="2">Specifies the reason the connection is being cleared (busy, for example).</td></tr>
</table>

Acknowledgement Model

OpenScape Voice supports the multi-step positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Response Parameters

Parameter Name	Supported	Comments
transferredCall	Y	Specifies the ConnectionID of the transferredTo device in the transferred call.

OpenScape Voice Operational Notes

- If the transferring device's Device ID refers to an extension number that appears on more than one device (multiple appearance), the Single Step Transfer Call service is only attempted on the device where the specified extension number is configured as its primary device, and only if no other appearances are active on that device.
- Prior to the Single Step Transfer, the local connection states of the active call at the transferring device and the transferred device must both be connected or a negative response is returned. This service is only supported for two-party calls; the active call cannot be a conference call or a consultation call.
- If the target is an invalid number, a Negative Ack is sent, not a positive response. For all the other cases, if a Positive Response is generated and the transfer does not take place, Failed and ConnectionCleared is generated
- In the case of network congestion condition in which the call cannot be transferred because there is no gateway available, a Positive Response is generated, but the transfer does not take place then Failed and ConnectionCleared events are generated.
- If the transferredTo device is call forwarded to another local party, the original SST target is indicated in the service response.
- After successful execution of the Single Step Transfer or manual blind transfer, the transferring device is placed into the blocked state (Failed Event) until the transfer is completed beyond the point of recall, then the blocked connection is cleared. Refer to Part II CSTA Call Scenarios for detailed event flows.
- If silent or seamless handover are possible, the Established Event will include an extended services permitted parameter to identify which service is available.
- This service is supported for ONS-IO.

Attention: One Number Service (ONS) is internal to Unify. Refer to Appendix F, "One Number Service (ONS)" for details.

- Improved Error Cause Codes on Immediate Recall

  This change impacts CSTA event flow for Blind Transfer (from phone), Single Step Transfer (via CSTA) and Silent Handover (via CSTA) for immediate recall scenarios. The change will impact the cause code presented in the ConnectionClearedEvent for the “pseudo” connectionID sent on the transferring device monitor. The following cause codes shall be presented:

  – **normalClearing** = call was successfully transferred to the requested device

  – **busy** = transferTo device is already active on a call and does not support call waiting

  – **destOutOfOrder** = could not reach the transferredTo device because it is out of service

  – **resourceNotAvailable** = transferredTo device is not able to terminate call due to resources limitations (e.g., Hunt Group queue lacks media server resources)

  – **notSupportedBearerService** = transferredTo device does not support compatible codec to handle the transferred device.

  – **networkCongestion** = No Call Admission Control (CAC) resources are available to connect to the transferredTo device

  – **networkNotObtainable** = could not reach the transferredTo device (e.g., WAN failure, SIP response timeout, unplugged device - i.e. registered but not reachable)

  – **reorderTone** = default cause code sent when a more specific cause cannot be provided

  – **incompatibleDestination** = transferred and transferredTo parties are the same (applies only to BT from phone)

  – **doNotDisturb** = transferredTo device has DND

  – **unknown** = UCE did not provide a release reason

  The event flow change described above applies to standard CSTA over TCP, CSTA over HTTP/SOAP and CSTA over SIP (OSMO).

Note: the ConnectionClearedEvent “pseudo” connection is independent of the Diverted and/or Established Events for the recall. The application shall not assume a fixed order of presentation of these events. The Established Event presented on the transferring device monitor will not be changed. The cause code in these events shall be “recall” or "recallBusy". If the application needs to track immediate recall and provide a recall reason then the information must be derived from the ConnectionClearedEvent on the “pseudo” connectionID as described above.

4.6.1.25 Transfer Call

The Transfer Call service transfers a call held at a device to an active call at the same device. The held and active calls at the transferring device merges into a new call. Also, the Connections of the held and active calls at the transferring device becomes Null and their ConnectionIDs ia released (that is, the transferring device is no longer involved with the call).

Service Request Parameters

Parameter Name	Supported	Comments	
heldCall	Y	Specifies the held connection.	
		Initial State	Supported
		Hold	Y
		Connected	N
activeCall	Y	Specifies the active connection.	
		Initial State	Supported
		Hold	Y
		Connected	N

Acknowledgement Model

OpenScape Voice supports the multi-step positive acknowledgement model.

Negative acknowledgement error values are defined in Section 2.6.2, “Negative Service Response”.

Response Parameters

Parameter Name	Supported	Comments
transferredCall	Y	Specifies the ConnectionID of the transferredTo device in the resulting call.

OpenScape Voice Operational Notes

- Connections on hold cannot be transferred. Conferences on hold cannot be transferred.
- Prior to the transfer, the transferring device must have a connection in the hold connection state, and a connection in the connected connection state, or a negative response is returned.
- Prior to the transfer, the connection at the transferred-to device must be either connected, hold, alerting or queued (group device only) or a negative response is returned.
- Prior to the transfer, the connection at the transferred device must be connected or a negative response is returned.

- After the successful transfer, the connection at the transferred device is still connected and the connection at the transferred-to-device is connected or alerting.
- If the transferring device's Device ID refers to an extension number that appears on more than one device (multiple appearances), the Transfer Call service is attempted only on the device where the specified extension number is configured as its primary device, and only if no other appearances are active on that device.
- If the transferred-to device was call forwarded to-another party, the forwarded-to-party is indicated in the service response.
- This service is supported for ONS-IO.

Attention: One Number Service (ONS) is internal to Unify. Refer to Appendix F, "One Number Service (ONS)" for details.

4.6.2 Events

OpenScape Voice supports servicesPermitted parameter as defined in Section 2.6.3, “Dynamic Feature Presentation (using servicesPermitted)”.

4.6.2.1 Bridged

Event Not Supported

The Bridged event indicates that an appearance at a shared bridged device configuration has been placed into an inactive mode (that is, queued state).

4.6.2.2 Call Cleared

Event Not Supported

The Call Cleared event is only provided for calls that are being call-type monitored. This event indicates that a call has been cleared and no longer exists within the switching sub-domain. A call is cleared when there is no longer any device associated with the call.

4.6.2.3 Conferenced

The Conferenced event indicates that the conferencing device has conferenced itself or another device with an existing call and that no devices have been removed from the resulting call.

Event Parameters

Parameter Name	Presented	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
primaryOldCall	Y	The switching function provides the local view option. The primaryOldCall specifies first active call leg on monitored device
secondaryOldCall	Y	The switching function provides the local view option. The secondaryOldCall specifies the second active call on the monitored device.
conferencingDevice	Y	Specifies the device ID of the conferencing device. **Note:** OpenScape Voice does not apply privacy restrictions to this parameter. Applications shall never display this parameter to other conference members.

Parameter Name	Presented	Comments
addedParty	Y	The addedParty specifies the device ID of the device that belongs to the active call of the conference. The device ID may be represented by Switching Function Representation or Device Number or Not Known.
conferenceConnections	Y	The conferenceConnections parameter is a list that contains the new ConnectionIDs and the old ConnectionIDs of the conference and for externally located devices the associated Network Interface DeviceID. OpenScape Voice provides connectionList with the following parameters: • newConnection (conference connection) • endpoint (deviceID only, in GNF or FQPN when available) • resultingConnectionInformation for: **mediaSessionInfo:** – Voice – Image **flowDirection** – Transmit - e.g. one-way speaker call – Receive - e.g. silent monitor service – Transmit&Receive - e.g, barge-in, whisper **Note:** numberOfChanges is not supported
localConnectionInfo	Y	Specifies the local connection state of the device associated with the Monitor Cross Reference ID. • Conferencing device – Connected • Other devices - See Table 17-148 in ECMA 269
cause	Y	The cause parameter specifies the reason for the event.
servicesPermitted	Y	Specifies a list of the call control services that can be applied to the local connection.
mediaCallCharacteristics	Y	This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. Supported value: • Voice
callLinkData	N	Specifies the correction data associated with the call.
callCharacteristics	Y	Specifies the characteristics that the switching function reports using the callCharacteristics parameter. Refer to callCharacteristicsSupported parameter description in the Section 4.1.1.5, “Get Switching Function Devices”
privateData	Y	Non-standardized information.

Event Causes

Cause Code	Conferenced Event	Example Reasons for Cause Code
Active Participation	Yes	Join Call, Silent Monitoring
Conference	Yes	Conference Call, Conference
Join Call	Yes	Join Call
Network Signal	No	External Call
Normal	No	Conference
Override	No	Intrude Call
Silent Participation	Yes	Join Call, Silent Monitoring

Miscellaneous Characteristics

- Local view shall be provided.

OpenScape Voice Operational Notes

- CSTA Subscriber must be provisioned for Large Conference Service.
- Event is not provided for control of SIP local device conferencing. Local SIP conference results in two connected calls.
- Silent Participation cause code requires special handling by application to ensure that devices involved in silent monitoring are not aware of joining device.
- There are cases where multiple external deviceIDs exist in the conferenceConnection list. This is non-standard CSTA implementation. Repeated ConnectionIDs in a conference can not be cleared via CSTA clear connection service.



4.6.2.4 Connection Cleared

The Connection Cleared event indicates that a single device has disconnected or dropped out of a call.

Event Parameters

Parameter Name	Presented	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
droppedConnection	Y	Specifies the connection of the device that was dropped from the call.
releasingDevice	Y	Specifies the device that dropped from the call. When sent on an NID monitor for an inbound call there are two cases: • Network Device releases: One event is sent with the CSTA endpoint deviceID • Called device releases: Two events are sent; one event for the called device as releasing and a second with the CSTA endpoint deviceID.
localConnection	Y	Specifies the local connection state of the device associated with the Monitor Cross Reference ID. The following are the states of the devices in the call: • Clearing device: Null • All other left in the call: Unaffected.
cause	Y	The cause parameter specifies the reason for the event.
servicesPermitted	Y	Specifies a list of the call control services that can be applied to the local connection.
mediaCallCharacteristics	Y	This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. Supported values: • Voice • Image **Note:** All media associated with a connection are considered cleared by OpenScape Voice.
callCharacteristics	Y	Specifies the characteristics that the switching function reports using the callCharacteristics parameter. Refer to callCharacteristicsSupported parameter description in the Section 4.1.1.3, “Get Switching Function Capabilities”
droppedConnectionInfo	N	Specifies the connection information associated with the dropped connection.
callLinkageData	N	Specifies the correlator data associated with the call.
privateData	Y	Non-standardized information.

Event Causes

Cause Code	Clear Connection	Example Reasons for Cause Code
Alert Time	N	Make Predictive Call
Busy	Y	Connection Failure
Busy Overflow	N	Clear Connection, Clear Call, Connection Failure
Calendar Overflow	N	Clear Connection, Clear Call, Connection failure
Call Back	Y	Call Back Call-Related, Call Back Message Call-Related
Call Cancelled	N	Clear Connection, Clear Call, Connection Failure
Call Not Answered	Y	Clear Connection, Clear Call, Connection failure
Capacity Overflow	N	Clear Connection, Clear Call, Connection Failure
Destination Detected	N	Make Predictive Call
Destination Not Obtainable	N	Connection Failure
Destination Out of Order	N	Connection Failure
Do Not Disturb	N	Do Not Disturb
Key Operation	Y	Bridged Call
Incompatible Destination[1]	N	Connection Failure
Invalid Account Code	N	Connection Failure
Invalid Number Format	N	Connection Failure
Maintenance	N	Connection Failure
Network Congestion	N	External Call
Network Not Obtainable	N	External Call
Network Out of Order	N	Connection Failure
Network Signal	N	Clear Connection, Clear Call, Connection Failure, External Call
Normal Clearing	Y	Clear Connection, Clear Call, Any feature
Not Available Bearer Service	N	Connection Failure
Not Supported Bearer Service	N	Connection Failure
Number Changed	N	Connection Failure
Number Unallocated	N	Connection Failure
Override	N	Connection Failure
Queue Cleared	N	ACD
Path Replacement	N	Network Feature

Cause Code	Clear Connection	Example Reasons for Cause Code
Queue Time Overflow	N	Clear Connection, Clear Call, Connection Failure
Reorder Tone	Y	Connection Failure
Resource not Available[2]	N	Connection Failure (generic – transient)
Selected Trunk Busy	N	Connection Failure
Trunks Busy	N	Connection Failure
Unauthorized Bearer Service	N	Connection Failure
Unknown Overflow	N	Clear Connection, Clear Call, Connection Failure

[1] OSCAR PROPOSAL: This cause shall be used for generic persistence connection failures unless an OSCAR specified cause code can be used.
[2] OSCAR PROPOSAL: This cause shall be used for generic transient connection failures unless an OSCAR specified cause code can be used.

OpenScape Voice Operational Notes

The connection cleared event is also provided when a device is removed from a call due to the call being transferred or diverted to another device. (The Transferred and Diverted events are used in these cases.)

4.6.2.5 Delivered

The Delivered event indicates that a call is being presented to a device in either the Ringing or Entering Distribution modes of the alerting state.

Event Parameters

Parameter Name	Presented	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
connection	Y	Specifies the connection that is alerting.
alertingDevice	Y	Specifies the device that is alerting.
callingDevice	Y	Specifies the device that is calling.
calledDevice *	Y	Specifies the originally called device.
lastRedirectionDevice	Y	Specifies the previously known redirected from device.
originatingNIDConnection	N	Specifies the connection of the Network Interface Device (NID) that the call originated from.
localConnectionInfo	Y	Specifies the local connection state of the device associated with the Monitor Cross Reference ID. The following are the states of the devices in the call. Alerting device: Alerting Calling device: is unaffected
cause	Y	The cause parameter specifies the reason for the event.
servicesPermitted	Y	Specifies a list of the call control services that can be applied to the local connection.
networkCallingDevice	N	Specifies the original calling device information provided by the network for external incoming calls.
networkCalledDevice	N	Specifies the original called device information provided by the network for external incoming calls.
associatedCallingDevice	Y	Specifies the Network Interface Device associated with the calling device if the call is an external incoming call. This parameter is mandatory for all external incoming calls and is not provided otherwise. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.

Parameter Name	Presented	Comments
associatedCalledDevice	Y	For outgoing external calls, this parameter specifies the Network Interface Device associated with the called device. For incoming external calls, this parameter specifies a device within the switching sub-domain associated with the originally called device. This parameter is mandatory for all external outgoing calls and it is optional for external incoming calls. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.
callCharacteristics	Y	Specifies the characteristics that the switching function reports using the callCharacteristics parameter. Refer to callCharacteristicsSupported parameter description in theSection 4.1.1.3, “Get Switching Function Capabilities”.
connectionInfo	Y	Specifies the connection information associated with the alerting connection. OpenScape Voice V9 presents the flowDirection information (e.g. 1-way speaker call): • Transmit&Receive • Receive • Transmit
mediaCallCharacteristics	Y	This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. Supported values: • Voice • Image **Note:** All media associated with a connection are considered delivered by OpenScape Voice.
callLinkData	N	Specifies the correlator data associated with the call.
subjectOfCall	N	Specifies the subject/intent of the call.
messageInfo	N	Specifies the contents of message information associated with a call.
privateData	Y	Non-standardized information.

* If the connection was redirected from a virtual subscriber (Remote Call Forwarding - RCF) device then the calledDevice will be presented as virtual subscriber DN. Refer to MLHG and RCF event flows in CSTA Call Scenarios for additional details.

Event Causes

Cause Code	Delivered Event	Example Reasons for Cause Code
ACD Busy	N	ACD, Consultation Call
ACD Forward	N	ACD
ACD Saturated	N	ACD, Consultation Call
Call Back	Y	Call Back Call (non-Call Related)
Call Forward	Y	Call Forwarding (generic / not specified)
Call Forward - Busy	Y	Call Forwarding (Busy)
Call Forward - Immediate	Y	Call Forwarding (Immediate)
Call Forward - No Answer	Y	Call Forwarding (No Answer)
Conference	N	Meet-Me Conference
Distributed	Y	ACD
Entering Distribution	Y	ACD
Key Operation	N	Multiple Appearance
Multiple Alerting	N	Multiple Alerting – Not yet supported
Network Signal	Y	External Calls
New Call	Y	Any feature (not already addressed by a specific event cause)
No Available Agents	N	ACD, Forwarding
Normal	N	Any feature (not supported, see new call)
Overflow	N	ACD (see call forward)
Override	N	Intrude Call
Recall	Y	Recall (generic / not specific)
Recall - Busy	Y	Recall
Recall - No Answer	Y	Recall
Recall - Forwarded	Y	Recall
Recall - Resources Not Available	Y	Recall
Redirected	Y	Deflect Call
Remains in Queue	N	ACD, Queuing, Single Step Conference, Join Call
Resources not available	N	ACD, Forwarding
Single Step Transfer	Y	Single Step Transfer
Single Step Conference	N	Single Step Conference
Timeout	N	External Calls

OpenScape Voice Operational Note

If a Group Device is provisioned for Queuing, OpenScape Voice sends a Delivered event before the Queued Event. See *Group Device queuing example flows in PART II - CSTA Call Scenarios*.

4.6.2.6 Dialed Digits

Event Not Supported

The Digits Dialed event indicates that a call or feature is being attempted from a device and that a portion of the dialing sequence has been completed. It implies that only part of the input activity is complete, and that more of the dialing sequence needs to be supplied before the entire input activity is complete. After the entire dialing sequence is complete, the Originated event is generated in the case where a call is being attempted.

4.6.2.7 Diverted

The Diverted event indicates that a call has been diverted from a device and that the call is no longer present at the device.

Event Parameters

Parameter Name	Presented	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
connection	Y	Specifies the connection that was diverted.
divertingDevice	Y	Specifies the device from which the call was diverted.
newDestination	Y	Specifies the device to which the call was diverted.
callingDevice	Y	Specifies the device remaining in the call with the newDestination
calledDevice*	Y	The calledDevice is either the originally dialed digits or the internal representation of the originally dialed number (after digit translation) or the DNIS in case of an incoming call. This parameter is only provided in case of Call Forward Immediate.
lastRedirectionDevice	Y	Specifies the previously known redirected from device
localConnectionInfo	Y	Specifies the local connection state of the device associated with the Monitor Cross Reference ID. The following are the states of the devices in the call. • Diverting device: Null • Others left in the call: are unaffected
cause	Y	The cause parameter specifies the reason for the event.
servicesPermitted	Y	Specifies a list of the call control services that can be applied to the local connection.
mediaCallCharacteristics	Y	This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. Supported values: • Voice • Image **Note:** All media associated with a connection are considered diverted by OpenScape Voice.

Parameter Name	Presented	Comments
callCharacteristics	Y	Specifies the characteristics that the switching function reports using the callCharacteristics parameter. Refer to callCharacteristicsSupported parameter description in Section 4.1.1.3, “Get Switching Function Capabilities”.
connectionInfo	N	Specifies the connection information associated with the diverted connection.
networkCallingDevice	N	Specifies the original calling device information provided by the network for external incoming calls.
networkCalledDevice	Y	Specifies the original called device information provided by the network for external incoming calls.
associatedCallingDevice	Y	Specifies the Network Interface Device associated with the calling device if the call is an external incoming call. This parameter is mandatory for all external incoming calls and shall not be provided otherwise. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.
associatedCalledDevice	Y	For outgoing external calls, this parameter specifies the Network Interface Device associated with the called device. For incoming external calls, this parameter specifies a device within the switching sub-domain associated with the originally called device. This parameter is mandatory for all external outgoing calls and it is optional for external incoming calls. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.
callLinkDataList	N	Specifies the correlator data associated with the call.
subjectOfCall	N	Specifies the subject/intent of the call.
messageInfo	N	Specifies the contents of message information associated with a call.
privateData	Y	Non-standardized information.

* If the connection was redirected from a virtual subscriber (Remote Call Forwarding - RCF) device then the calledDevice will be presented as virtual subscriber DN. Refer to MLHG and RCF event flows in CSTA Call Scenarios for additional details.

Event Causes

Highlighted cause codes need to be re-evaluated after OSCAR CSTA Interface Specification is released.

Cause Code	Diverted Event	Example Reasons for Cause Code
ACD Forward	N	ACD

Cause Code	Diverted Event	Example Reasons for Cause Code
Busy Overflow	N	Call Forwarding
Calendar Overflow	N	Call Forwarding
Call Forward	Y	Call Forwarding
Call Forward - Busy	Y	Call Forwarding
Call Forward - Immediate	Y	Call Forwarding
Call Forward - No Answer	Y	Call Forwarding
Call Not Answered	Y	Recall (Group Device (MLHG) - No answer advance)
Call Pickup	Y	Directed Pickup Call, Group Pickup Call
Capacity Overflow	N	Call Forwarding
Distributed	Y	ACD
Do Not Disturb	N	Call Forwarding (CFW DND not supported)
No Available Agents	Y	ACD
Normal	Y	Any Feature
Overflow	Y	ACD
Park	Y	Park Call (Future) Park to Server
Path Replacement	N	Network Feature
Queue Time Overflow	Y	Call Forwarding (Should be changed to Call Forward)
Queue Night Service	Y	CQ to change Call Forward.
Recall	Y	Recall
Recall – Busy	Y	Recall
Recall – No Answer	Y	Recall
Recall - Forwarded	Y	Recall (Intercept)
Recall – Resources Not Available	N	Recall
Redirected	Y	Deflect Call
Resources Not Available	Y	ACD, Forwarding (Night Service DN unknown)
Unknown Overflow	N	Call Forwarding

OpenScape Voice Operational Notes

- Diverted events are sent to all affected parties in a call. For the Deflect service, for example, there are 2 Diverted events generated—one for the originally called party and one for the calling party.
- Diverted events reflect interworking with SIPQ Call Forwarding scenarios (Unconditional, Busy and No Reply).

4.6.2.8 Established

The Established event indicates that a call has been answered at a device or that a call has been connected to a device.

Event Parameters

Parameter Name	Presented	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
established**C**onnection	Y	Specifies the connection that was connected.
answeringDevice	Y	Specifies the device that connected into the call.
callingDevice	Y	Specifies the calling device
calledDevice *	Y	Specifies the called device
lastRedirectionDevice	Y	Specifies the previously known redirected from device
originatingNIDConnection	N	Specifies the connection of the Network Interface Device (NID) that the call originated from.
localConnectionInfo	Y	Specifies the local connection state of the device associated with the Monitor Cross Reference ID. The following are the states for: • Answering device: Connected • Calling device: unaffected and never Null
cause	Y	The cause parameter specifies the reason for the event.
servicesPermitted	Y	Specifies a list of the call control services that can be applied to the local connection.
networkCallingDevice	N	Specifies the original calling device information provided by the network for external incoming calls.
networkCalledDevice	Y	Specifies the original called device information provided by the network for external incoming calls.

Parameter Name	Presented	Comments
associatedCallingDevice	Y	Specifies the Network Interface Device associated with the calling device if the call is an external incoming call. This parameter is mandatory for all external incoming calls and shall not be provided otherwise. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.
associatedCalledDevice	Y	For outgoing external calls, this parameter specifies the Network Interface Device associated with the called device. For incoming external calls, this parameter specifies a device within the switching sub-domain associated with the originally called device. This parameter is mandatory for all external outgoing calls and it is optional for external incoming calls. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.
mediaCallCharacteristics	Y	This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. Supported values: • Voice • Image **Note:** All media associated with a connection are considered connected by OpenScape Voice.
callCharacteristics	Y	Specifies the characteristics that the switching function reports using the callCharacteristics parameter. Refer to callCharacteristicsSupported parameter description in the Section 4.1.1.3, “Get Switching Function Capabilities”
establishedConnectionInfo	Y	Specifies the connection information associated with the established connection. Supported values for mediaSessionInfo: – Voice – Image Supported values for flowDirection – Transmit - e.g. one-way speaker call – Receive - e.g. silent monitor service – Transmit&Receive - e.g, barge-in, whisper **Note:** numberOfChanges is not supported

Parameter Name	Presented	Comments
callLinkData	N	Specifies the correlator data associated with the call.
subjectOfCall	N	Specifies the subject/intent of the call.
messageInfo	N	Specifies the contents of message information associated with a call.
privateData	Y	• Non-standardized information. • Includes extendedServicesPermitted for silent and seamless handover.

* If the connection was redirected from a virtual subscriber (Remote Call Forwarding - RCF) device then the calledDevice will be presented as virtual subscriber DN. Refer to MLHG and RCF event flows in CSTA Call Scenarios for additional details.

Event Causes

Event Causes	Established Event	Example Reasons for Cause Code
ACD Forward	N	ACD
Alternate	Y	Alternate Call
Call Back	Y	Call Back Call-Related
Call Forward	Y	Call Forwarding
Call Forward - Busy	Y	Call Forwarding
Call Forward - Immediate	Y	Call Forwarding
Call Forward - No Answer	Y	Call Forwarding
Call Pickup	Y	Directed Pickup Call, Group Pickup Call
Conference	Y	Conference
Distributed	Y	ACD, Routing Services
Intrude	N	Intrude Call
Key Operation	N	Multiple Appearance
Network Signal	N	External Calls
New Call	Y	Any Feature
No Available Agents	Y	ACD, Forwarding
Normal	Y	Any Feature
Overflow	Y	ACD (Group Device (MLHG) – call distributed to the overflow DN)
Override	N	Intrude Call
Recall	Y	Recall
Recall – Busy	Y	Recall
Recall – No Answer	Y	Recall
Recall - Forwarded	Y	Recall
Recall – Resources Not Available	?	Recall

Event Causes	Established Event	Example Reasons for Cause Code
Redirected	Y	Deflect Call
Remains in Queue	Y	ACD
Resources Not Available	Y	ACD, Forwarding
Single Step Conference	N	Single Step Conference
Single Step Transfer	Y	Single Step Transfer
Timeout	N	External Call

OpenScape Voice Operational Note

EstablishedConnectionInfo is used in silent monitoring and Join call event flows.

4.6.2.9 Failed

The Failed event indicates that a call cannot be completed or a connection has entered the Fail state for any reasons.

Event Parameters

Parameter Name	Presented	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
failedConnection	Y	Specifies the connection that failed.
failingDevice	Y	Specifies the device that failed.
callingDevice	Y	Specifies the calling device
calledDevice *	Y	Specifies the called device
lastRedirectionDevice	Y	Specifies the previously known redirected from device
originatingNIDConnection	N	Specifies the connection of the Network Interface Device (NID) that the call originated from.
localConnectionInfo	Y	Specifies the local connection state of the device associated with the Monitor Cross Reference ID. The following are the states of the devices in the call. For the failing device: Fail For the other devices left in the call: (unaffected)
cause	Y	The cause parameter specifies the reason for the event.
servicesPermitted	Y	Specifies a list of the call control services that can be applied to the local connection.
networkCallingDevice	Y	Specifies the original calling device information provided by the network for external incoming calls.
networkCalledDevice	N	Specifies the original called device information provided by the network for external incoming calls.

Parameter Name	Presented	Comments
associatedCallingDevice	Y	Specifies the Network Interface Device associated with the calling device if the call is an external incoming call. This parameter is mandatory for all external incoming calls and shall not be provided otherwise. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.
associatedCalledDevice	Y	For outgoing external calls, this parameter specifies the Network Interface Device associated with the called device. For incoming external calls, this parameter specifies a device within the switching sub-domain associated with the originally called device. This parameter is mandatory for all external outgoing calls and it is optional for external incoming calls. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.
mediaCallCharacteristics	Y	This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. Supported values: • Voice • Image **Note:** All media associated with a connection are considered failed by OpenScape Voice.
callCharacteristics	Y	Specifies the characteristics that the switching function reports using the callCharacteristics parameter. Refer to callCharacteristicsSupported parameter description in Section 4.1.1.3, “Get Switching Function Capabilities”.
failedConnectionInfo	Y	Specifies the connection information associated with the failed connection. Supported values for mediaSessionInfo: – Voice – Image Supported values for flowDirection – Transmit - e.g. one-way speaker call – Receive - e.g. silent monitor service – Transmit&Receive - e.g, barge-in, whisper **Note:** numberOfChanges is not supported

Parameter Name	Presented	Comments
callLinkData	N	Specifies the correlator data associated with the call.
subjectOfCall	N	Specifies the subject/intent of the call.
messageInfo	N	Specifies the contents of message information associated with a call.
privateData	Y	Non-standardized information.

* If the connection was redirected from a virtual subscriber (Remote Call Forwarding - RCF) device then the calledDevice will be presented as virtual subscriber DN. Refer to MLHG and RCF event flows in CSTA Call Scenarios for additional details.

Event Causes

Supported Event Causes	Failed Event	Example Reasons for Cause Code
Blocked	Yes	After Single Step Transfer - transferring device is temporarily blocked until the transfer completes beyond the possibility of recall.
Busy	Yes	Connection failed due to busy device.
Call Cancelled	Yes	Calling party abandons (hangs up) after call is diverted from queue to agent before agent is alerted.
Call Not Answered	No	Used in Diverted Event for Recall on No Answer
Destination Not Obtainable	Yes	Media server request for an unregistered subscriber.
Destination Out of Order	Yes	Device is not registered with OpenScape Voice registrar
Do Not Disturb	Yes	Do Not Disturb subscriber feature active on called device
Invalid Number Format	Yes	Digit translation errors
Key Operation	No	Used in Conferenced, Held and Retrieved events in Bridged Call use cases
Network Congestion	Yes	CAC - Resource Management Error - no subscriber re-routing
Network Not Obtainable	Yes	CAC - Subscriber re-routing
Network Signal	No	Network interface device (NNI) used on call
Normal	Yes	Used when a more specific cause can not be provided.
Redirected	Yes*	*While not supported by ECMA this cause is in an ACD event flow when a call distributed to an agent whose phone has local call forwarding set. In this case the call is pulled back to the Hunt Group for re-hunting or application distribution.
Reorder Tone	Yes	Number translation error.

Supported Event Causes	Failed Event	Example Reasons for Cause Code
Resources Not Available	Yes	The following cases apply: • In Large Conference if exceeded the Number of members • Accept to an Unavailable subscriber • Forwarding due to unregistered/unreacheable devices • Some Busy-Recall scenarios • Problems with internal resources (e.g. SSAL memory slot not available) • Failed to connect to Media Server (Large Conference)

OpenScape Voice Operational Notes

A complete connection identifier is provided as the failedConnection parameter and the Failed event is reported to all active monitors associated with the call. The switching function does not provide userData with this event, although it is required by the standard.

4.6.2.10 Held

The Held event indicates that a call has been placed on hold.

Event Parameters

Parameter Name	Presented	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
heldConnection	Y	Specifies the connection at which the hold was activated.
holdingDevice	Y	Specifies the device at which hold was activated.
localConnectionInfo	Y	Specifies the local connection state of the device associated with the Monitor Cross Reference ID. The following are the states of the devices in the call: • Holding device: Hold • Other left in the call: are unaffected.
cause	Y	The cause parameter specifies the reason for the event.
servicesPermitted	Y	Specifies a list of the call control services that can be applied to the local connection.
mediaCallCharacteristics	Y	This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. Supported values for mediaClass: • Voice • Image --- **Note:** All media associated with a connection are considered held by OpenScape Voice. ---
callCharacteristics	Y	Specifies the characteristics that the switching function reports using the callCharacteristics parameter. Refer to callCharacteristicsSupported parameter description in Section 4.1.1.3, “Get Switching Function Capabilities”.
heldConnectionInfo	Y	Specifies the connection information associated with the held connection. Supported values for mediaSessionInfo: – Voice – Image Supported values for flowDirection – Transmit - e.g. one-way speaker call – Receive - e.g. silent monitor service – Transmit&Receive - e.g, barge-in, whisper numberOfChanges is not supported
callLinkData	N	Specifies the correlator data associated with the call.

Parameter Name	Presented	Comments
privateData	Y	Non-standardized information.

Event Causes

Highlighted cause codes need to be re-evaluated after OSCAR CSTA Interface Specification is released.

Event Causes	Held Event	Example Reasons for Cause Code
Alternate	Y	Alternate Call
Conference	Y	Consultation Call with a consultOptions of ConferenceOnly
Consultation	Y	Consultation Call
Key Operation	Y	Bridged Call
Intrude	N	Intrude Call
Maintenance	Y	Maintenance
Network Signal	N	External Call
Normal	Y	Any Feature
Recall	Y	Recall
Suspend	N	Call Clearing
Transfer	Y	Consultation Call with a consultOptions of Transfer Only

OpenScape Voice Operational Notes

None

4.6.2.11 Network Capabilities Changed

Event Not Supported

The Network Capabilities Changed event indicates that a situation occurred during a call's progress in a public or private network that modifies its signaling capability (that is, inter-networking). It does not indicate a change in the connection state of the Network Interface Device (for example, trunk, CO Line) through which the call has accessed the network.

4.6.2.12 Network Reached

The Network Reached event indicates that a call has cut through the switching sub-domain boundary to another network; that is, has reached and engaged a Network Interface Device (for example, trunk, CO Line). This event indicates that there may be a reduced level of event reporting and possibly no additional device feedback, except connection/call clearing, provided for this device in the call due to a lack of network signaling. The level of signaling provided by the network may be indicated by the networkCapability parameter.

Event Parameters

Parameter Name	Presented	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
outboundConnection	Y	Specifies the outbound connection associated with the call that is leaving the switching sub-domain.
networkInterfaceUsed	Y	Specifies the Network Interface Device that was selected. Parameter is always set to “notKnown”
callingDevice	Y	Specifies the calling device
calledDevice *	Y	The calledDevice is either the originally dialed digits or the internal representation of the originally dialed number (after digit translation) or the DNIS in case of an incoming call.
lastRedirectionDevice	Y	Specifies the previously known redirected from device
originatingNIDConnection	N	Specifies the connection of the Network Interface Device (NID) that the call originated from.
localConnectionInfo	Y	Specifies the local connection state of the device associated with the Monitor Cross Reference ID. The following are the states of the devices in the call: • Network Interface Device: Connected • Other devices left in the call: are unaffected
networkCapability	N	Specifies the type of network reached and the Call Control events supported by the network.
cause	Y	The cause parameter specifies the reason for the event.
servicesPermitted	Y	Specifies a list of the call control services that can be applied to the local connection.

Parameter Name	Presented	Comments
mediaCallCharacteristics	Y	This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. Supported values for mediaSessionInfo: – Voice – Image Supported values for flowDirection – Transmit - e.g. one-way speaker call – Receive - e.g. silent monitor service – Transmit&Receive - e.g, barge-in, whisper **Note:** numberOfChanges is not supported **Note:** Only specified media is applicable and is determined by the capabilities for the gateway interface.
callCharacteristics	Y	Specifies the characteristics that the switching function reports using the callCharacteristics parameter. Refer to callCharacteristicsSupported parameter description in Section 4.1.1.3, “Get Switching Function Capabilities”.
outboundConnectionInfo	N	Specifies the connection information associated with the outboundConnection connection.
networkCallingDevice	N	Specifies the original calling device information provided by the network for external incoming calls.
networkCalledDevice	Y	Specifies the original called device information provided by the network for external incoming calls.
associatedCallingDevice	Y	Specifies the Network Interface Device associated with the calling device if the call is an external incoming call. This parameter is mandatory for all external incoming calls and shall not be provided otherwise. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.

Parameter Name	Presented	Comments
associatedCalledDevice	N	For outgoing external calls, this parameter specifies the Network Interface Device associated with the called device. For incoming external calls, this parameter specifies a device within the switching sub-domain associated with the originally called device. This parameter is mandatory for all external outgoing calls and it is optional for external incoming calls. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.
outboundConnectionInfo	N	Specifies the connection information associated with the outbound connection.
callLinkData	N	Specifies the correlator data associated with the call.
privateData	Y	Non-standardized information.

* If the connection was redirected from a virtual subscriber (Remote Call Forwarding - RCF) device then the calledDevice will be presented as virtual subscriber DN. Refer to MLHG and RCF event flows in CSTA Call Scenarios for additional details.

Event Causes

Event Causes	Network Reached Even	Example Reasons for Cause Code
ACD Forward	N	ACD
Call Forward	Y	Forwarding
Call Forward - Busy	Y	Forwarding
Call Forward - Immediate	Y	Forwarding
Call Forward - No Answer	Y	Forwarding
Conference	Y	Any Conference Feature
Distributed	Y	ACD
No Available Agents	?	ACD, distribution
Normal	Y	Any Features
Overflow	Y	ACD
Redirected	Y	Deflect Call
Resources Not Available	Y	ACD, distributed
Transfer	Y	Transfer

OpenScape Voice Operational Notes

None

4.6.2.13 Offered

The Offered event indicates that the connection is in the Offered mode of the Alerting state. This indicates that a call is in a pre-delivery state.

Event Parameters

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
offeredConnection	Y	Specifies the connection that is altering (offer mode). **Note:** When presented on a Network Interface device monitor the connection ID is non-standard. In order to ensure that the connectionID presented on the NID monitor the deviceID portion of the connectionIDs is not presented in Device Number (DeN) format, but instead is presented in SFR format. The DN portion of the deviceID shall be the FQDN of the external calling device. In addition a new NM field tag “nid” will include the string representation of the network interface deviceID. Also since the DN is external the “uid” tag is also included in the NM field. For example: X1 calls D2 over NID1 creating connection C1. The Calling deviceID SFR would be presented as: N<X1>;uid=C1;nid=NID1.
offeredDevice	Y	Specifies the device that is alerting **Note:** Network Interface devices are presented in Device Number (DeN) format. See Section 2.6.1.3, “Device Number (DeN) Format”.
callingDevice	Y	Specifies the calling device.
calledDevice *	Y	Specifies the originally called device.
lastRedirectionDevice	Y	Specifies the previously known redirected from device.
localConnectionInfo	Y	Specifies the local connection state of the device associated with the Monitor Cross Reference ID. The following are the states of the devices in the call. • For the alerting device: Alerting • For the calling device: (unaffected)
cause	Y	The cause parameter specifies the reason for the event.

Parameter Name	Supported	Comments
servicesPermitted	Y	Specifies a list of the call control services that can be applied to the local connection.
associatedCallingDevice	Y	Specifies the Network Interface Device associated with the calling device if the call is an external incoming call. This parameter is mandatory for all external incoming calls and shall not be provided otherwise.
associatedCalledDevice	Y	For outgoing external calls, this parameter specifies the Network Interface Device associated with the called device. For incoming external calls, this parameter specifies a device within the switching sub-domain associated with the originally called device. This parameter is mandatory for all external outgoing calls and it is optional for external incoming calls.
networkCallingDevice	N	Specifies the original calling device information provided by the network for external incoming calls.
networkCalledDevice	Y	Specifies the original called device information provided by the network for external incoming calls.
associatedCallingDevice	Y	Specifies the Network Interface Device associated with the calling device if the call is an external incoming call. This parameter is mandatory for all external incoming calls and shall not be provided otherwise. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.
associatedCalledDevice	Y	For outgoing external calls, this parameter specifies the Network Interface Device associated with the called device. For incoming external calls, this parameter specifies a device within the switching sub-domain associated with the originally called device. This parameter is mandatory for all external outgoing calls and it is optional for external incoming calls. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.
mediaCallCharacteristics	Y	This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. Supported values: • Voice • Image

Parameter Name	Supported	Comments
callCharacteristics	Y	Specifies the characteristics that the switching function reports using the callCharacteristics parameter. Refer to callCharacteristicsSupported parameter description in tSection 4.1.1.3, “Get Switching Function Capabilities”.
offeredConnectionInfo	Y	Specifies the connection information associated with the alerting connection. Supported values for mediaSessionInfo: – Voice – Image Supported values for flowDirection – Transmit - e.g. one-way speaker call – Receive - e.g. silent monitor service – Transmit&Receive - e.g, barge-in, whisper **Note:** numberOfChanges is not supported
callLinkData	N	Specifies the correlator data associated with the call.
subjectOfCall	N	Specifies the subject/intent of the call.
messageInfo	N	Specifies the contents of message information associated with a call.

* If the connection was redirected from a virtual subscriber (Remote Call Forwarding - RCF) device then the calledDevice will be presented as virtual subscriber DN. Refer to MLHG and RCF event flows in CSTA Call Scenarios for additional details.

Event Causes

Highlighted cause codes need to be re-evaluated after OSCAR CSTA Interface Specification is released.

Event Causes	Supported	Example Reasons for Cause Code
ACD Busy	N	ACD, Consultation Call
ACD Forward	N	ACD
ACD Saturated	N	ACD, Consultation Call
Call Back	Y	Call Back Call-Related
Call Forward	Y	Call Forwarding
Call Forward—Busy	Y	Call Forwarding
Call Forward—Immediate	Y	Call Forwarding
Call Forward—No Answer	Y	Call Forwarding
Conference	Y	Any Conference Feature
Distributed	Y	ACD, Routing Services
Entering Distribution	Y	ACD, Routing Services
Key Operation	N	Multiple Appearance
Multiple Alerting	Y	Multiple Alerting (Simultaneous Ringing Service)

Event Causes	Supported	Example Reasons for Cause Code
Network Signal	N	
New Call	Y	Any Feature
No Available Agents	Y	ACD, Forwarding
Normal	Y	Any Feature
Overflow	Y	ACD
Override	N	Intrude Call
Recall	Y	Recall
Recall – Busy	Y	Recall
Recall - No Answer	N	Recall
Recall – Forwarded	Y	Recall
Recall - Resources Not Available	N	Recall
Redirected	Y	Deflect Call
Remains in Queue	Y	ACD, Queuing, Single Step Conference, Join Call
Resources Not Available	Y	ACD, Forwarding
Single Step Transfer	Y	Single Step Transfer
Single Step Conference	N	Single Step Conference
Timeout	N	External Calls

OpenScape Voice Operational Notes

1. This event is flowed by the OpenScape Voice when a call reaches a monitored ONS Subscriber. Internally, the call is in a suspended state waiting for the application to operate on the call via a Deflect Call, Accept Call or Clear Connection.
2. The Offered event will not be generated in the case of SS Transfer or Transfer. While generated for consultation call, it will not be generated for consultation transfer.
3. Offered is not supported for MLHG pilot since ONS-IO is currently not supported for MLHG pilots.
4. If the monitored ONS subscriber is involved in one or more calls the Offered event will still be generated.
5. The call forward-dependable or call forwarding-immediate switching function service will be applied after the Accept Call service but before the device is alerted.

6. The application may suppress ONS Inbound by filtering the Offered event at the time the monitor is started for an ONS subscriber DN. Inbound calls are treated as though the Offered Event times out immediately. The call proceeds to the registering device (if any) and may be affected by other feature interactions such as call forwarding.

Attention: One Number Service (ONS) is internal to Unify. Refer to Appendix F, “One Number Service (ONS)” for details.

4.6.2.14 Originated

The Originated event indicates that a call is being attempted from a device. It implies that input activity for the call is complete and that a call (rather than a feature) has been requested.

Event Parameters

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
originatedConnection	Y	Specifies the connection at which the call originated. **Note:** When presented on a Network Interface device monitor the connection ID is non-standard. In order to ensure that the connectionID presented on the NID monitor the deviceID portion of the connectionIDs is not presented in Device Number (DeN) format, but instead is presented in SFR format. The DN portion of the deviceID shall be the FQDN of the external calling device. In addition a new NM field tag “nid” will include the string representation of the network interface deviceID. Also since the DN is external the “uid” tag is also included in the NM field. For example: X1 calls D2 over NID1 creating connection C1. The Calling deviceID SFR would be presented as: N<X1>;uid=C1;nid=NID1.
callingDevice	Y	For internal or outgoing calls the callingDevice parameter includes the Switching Function Representation of the calling device, for incoming calls the callingDevice parameter may include the Automatic Number Identification (ANI)
calledDevice *	Y	The calledDevice is the originally dialed digits. For 1-way speaker broadcast calls it shall be the broadcast feature access code + Broadcast Group Number:
originatingDevice	Y	Specifies the initiating device. In the case of NID; For example: X1 (external) calls D2 on callID=C1. The originating deviceID SFR would be presented as: N<X1>;uid=C1
localConnectionInfo	Y	Specifies the local connection state of the device associated with the Monitor Cross Reference ID. The following are the states of the devices in the call. For the device initiating the call: Connected

Parameter Name	Supported	Comments
cause	Y	The cause parameter specifies the reason for the event.
servicesPermitted	Y	Specifies a list of the call control services that can be applied to the local connection.
networkCallingDevice	N	Specifies the original calling device information provided by the network for external incoming calls.
networkCalledDevice	Y	Specifies the original called device information provided by the network for external incoming calls.
associatedCallingDevice	Y	Specifies the Network Interface Device associated with the calling device if the call is an external incoming call. This parameter is mandatory for all external incoming calls and shall not be provided otherwise. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used. **Note:** Network Interface devices are presented in Device Number (DeN) format. See Section 2.6.1.3, “Device Number (DeN) Format”.
associatedCalledDevice	Y	For outgoing external calls, this parameter specifies the Network Interface Device associated with the called device. For incoming external calls, this parameter specifies a device within the switching sub-domain associated with the originally called device. This parameter is mandatory for all external outgoing calls and it is optional for external incoming calls. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.
mediaCallCharacteristics	Y	This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. Supported values: • Voice • Image **Note:** All media associated with a connection are considered originated for the call by OpenScape Voice.

Parameter Name	Supported	Comments
callCharacteristics	Y	Specifies the characteristics that the switching function reports using the callCharacteristics parameter. Refer to callCharacteristicsSupported parameter description in Section 4.1.1.3, "Get Switching Function Capabilities".
originatedConnectionInfo	Y	Specifies the connection information associated with the originated connection. Supported values for mediaSessionInfo: – Voice – Image Supported values for flowDirection – Transmit - e.g. one-way speaker call – Receive - e.g. silent monitor service – Transmit&Receive - e.g, barge-in, whisper **Note:** numberOfChanges is not supported
callLinkData	N	Specifies the correlator data associated with the call.
subjectOfCall	N	Specifies the subject/intent of the call.
messageInfo	N	Specifies the contents of message information associated with a call.
privateData	Y	Non-standardized information.

* If the connection was redirected from a virtual subscriber (Remote Call Forwarding - RCF) device then the calledDevice will be presented as virtual subscriber DN. Refer to MLHG and RCF event flows in CSTA Call Scenarios for additional details.

Event Causes

Highlighted cause codes need to be re-evaluated after OSCAR CSTA Interface Specification is released.

Event Causes	Supported	Example Reasons for Cause Code
Call Back	Y	Call Back Non-Call-Related
Conference	N	Consultation Call with a consultOptions of ConferenceOnly
Consultation	N	Consultation Call
Make Call	N	Make Call without
New Call	Y	Any Feature
Normal	Y	Any Feature, Make Call
Transfer	N	Consultation Call with a consultOptions of TransferOnly

OpenScape Voice Operational Note

The switching function does not provide the Originated event cases where a connection immediately reports failed.

4.6.2.15 Queued

The Queued event indicates that a call has been queued.

If enabled via CSTA interface option, an “All Members Busy (AMB)” notification is presented via a special non-call related Queued Event presented on monitored Hunt Group pilot.

Event Parameters

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
queuedConnection	Y	Specifies the queued connection.
queue	Y	Specifies the queue device. The device ID may be represented by Switching Function Representation or Device Number.
callingDevice	Y	For internal or outgoing calls the callingDevice parameter includes the Switching Function Representation of the calling device, for incoming calls the callingDevice parameter may include the Automatic Number Identification (ANI) If all members of a Hunt Group are busy then a special non-call related Queued Event may be presented with this parameter set to “notKnown”.
calledDevice *	Y	The calledDevice is either the originally dialed digits or the internal representation of the originally dialed number (after digit translation) or the DNIS in case of an incoming call.
lastRedirectionDevice	Y	Specifies the previously known redirected from device
localConnectionInfo	Y	Specifies the local connection state of the device associated with the Monitor Cross Reference ID. The following are the states of the devices in the call. For the queue device: Queued For the other devices left in the call: (unaffected)
cause	Y	The cause parameter specifies the reason for the event.
servicesPermitted	Y	Specifies a list of the call control services that can be applied to the local connection.
networkCallingDevice	N	Specifies the original calling device information provided by the network for external incoming calls.
networkCalledDevice	Y	Specifies the original called device information provided by the network for external incoming calls.

Parameter Name	Supported	Comments
associatedCallingDevice	Y	Specifies the Network Interface Device associated with the calling device if the call is an external incoming call. This parameter is mandatory for all external incoming calls and shall not be provided otherwise. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.
associatedCalledDevice	Y	For outgoing external calls, this parameter specifies the Network Interface Device associated with the called device. For incoming external calls, this parameter specifies a device within the switching sub-domain associated with the originally called device. This parameter is mandatory for all external outgoing calls and it is optional for external incoming calls. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.
mediaCallCharacteristics	Y	This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. Supported values: • Voice only
callCharacteristics	Y	Specifies the characteristics that the switching function reports using the callCharacteristics parameter. Refer to callCharacteristicsSupported parameter description inSection 4.1.1.3, “Get Switching Function Capabilities”
initiatedConnectionInfo	N	Specifies the connection information associated with the initiated connection.
callLinkData	N	Specifies the correlator data associated with the call.
subjectOfCall	N	Specifies the subject/intent of the call.
messageInfo	N	Specifies the contents of message information associated with a call.
servicesPermitted	Y	Specifies the services permitted for a queued call. In the case if the special Queued Event when all members are busy, there are no services permitted.
privateData	Y	Non-standardized information.

* If the connection was redirected from a virtual subscriber (Remote Call Forwarding - RCF) device then the calledDevice will be presented as virtual subscriber DN. Refer to MLHG and RCF event flows in CSTA Call Scenarios for additional details.

Event Causes

Highlighted cause codes need to be re-evaluated after OSCAR CSTA Interface Specification is released.

Cause Codes	Queued Event	Example Reasons for Cause Code
Busy	Y	Make Call, Consultation Call, Deflect, Single-Step Transfer Call, Single-Step Conference Call
Call Forward	Y	Call Forwarding
Call Forward—Busy	Y	Call Forwarding
Call Forward—Immediate	Y	Call Forwarding
Call Forward—No Answer	Y	Call Forwarding
Camp On	N	Camp On Call (no queued event supported for endpoints. All calls delivered)
Camp On Trunks	N	External Call, Camp On Call
Distribution Delay	Y	ACD, queuing (Call queued and no members in the group are busy or Working after call.)
Do Not Disturb	N	Forwarding
Multiple Queuing	N	Multiple Queuing
Network Congestion	Y	External Call
Network Not Obtainable	N	External Call
Network Signal	N	External Call
No Available Agents	Y	ACD
Normal	Y	Any feature
Overflow	Y	ACD, queuing (Call distributed from group device to overflow destination)
Park	Y	Park Call (Future) Park to Server
Recall	Y	Recall
Recall – Busy	N	Recall
Recall - No Answer	N	Recall
Recall – Forwarded	N	Recall
Recall - Resources Not Available	N	Recall
Redirected	Y	Deflect
Remains in Queue	Y	ACD, Queuing, Single Step Conference, Join Call
Resources Not Available	Y	ACD

Cause Codes	Queued Event	Example Reasons for Cause Code
Trunks Busy	N	External Call

OpenScape Voice Operational Notes

1. The queued event is currently only provided for Group Devices only, not Station Devices, for example, no call waiting.

2. A QueuedEvent is optionally presented when all members of the group are busy (AMB). The AMB event includes callingDevice = <notKnown />; cause = noAgentsAvailable and no servicesPermitted. The AMB condition is removed by a ConnectionClearedEvent on the connectionID specified in the QueuedEvent. AMB event XML example:

```
<?xml version="1.0" encoding="UTF-8"?>
<QueuedEvent xmlns="http://www.ecma-international.org/
standards/ecma-323/csta/ed4">
 <monitorCrossRefID>1158243216</monitorCrossRefID>
 <queuedConnection>
   <callID>FF00010000000000D8F60452CD150000</callID>
   <deviceID>+15619231416</deviceID>
 </queuedConnection>
 <queue>
   <deviceIdentifier>+15619231416</deviceIdentifier>
 </queue>
 <callingDevice>
   <notKnown />
 </callingDevice>
 <calledDevice>
   <deviceIdentifier>+15619231416</deviceIdentifier>
 </calledDevice></font>
 <lastRedirectionDevice>
   <notRequired />
 </lastRedirectionDevice>
 <localConnectionInfo>queued</localConnectionInfo>
 <cause>noAvailableAgents</cause>
 <servicesPermitted>
   <callControlServices>
   </callControlServices>
   <callAssociatedServices />
   <mediaAttachementServices />
   <routeingServices />
   <voiceServices />
 </servicesPermitted>
<mediaCallCharacteristics>
```

4.6.2.16 Retrieved

The Retrieved event indicates that a previously held call has been retrieved.

Event Parameters

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
retrievedConnection	Y	Specifies the connection at which hold was deactivated.
retrievingDevice	Y	Specifies the device at which hold was deactivated.
localConnectionInfo	Y	Specifies the local connection state of the device associated with the Monitor Cross Reference ID. The following are the states of the devices in the call. For the retrieving device: Connected For the other devices left in the call: (unaffected)
cause	Y	The cause parameter specifies the reason for the event.
servicesPermitted	Y	Specifies a list of the call control services that can be applied to the local connection.
networkCallingDevice	N	Specifies the original calling device information provided by the network for external incoming calls.
networkCalledDevice	Y	Specifies the original called device information provided by the network for external incoming calls.
associatedCallingDevice	Y	Specifies the Network Interface Device associated with the calling device if the call is an external incoming call. This parameter is mandatory for all external incoming calls and shall not be provided otherwise. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.
associatedCalledDevice	Y	For outgoing external calls, this parameter specifies the Network Interface Device associated with the called device. For incoming external calls, this parameter specifies a device within the switching sub-domain associated with the originally called device. This parameter is mandatory for all external outgoing calls and it is optional for external incoming calls. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.

Parameter Name	Supported	Comments
mediaCallCharacteristics	Y	This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. Supported values: • Voice • Image **Note:** All media associated with a connection are considered retrieved by OpenScape Voice.
callCharacteristics	Y	Specifies the characteristics that the switching function reports using the callCharacteristics parameter. Refer to callCharacteristicsSupported parameter description in Section 4.1.1.3, "Get Switching Function Capabilities".
retrievedConnectionInfo	Y	Specifies the connection information associated with the alerting connection. Specifies the connection information associated with the originated connection. Supported values for mediaSessionInfo: – Voice – Image Supported values for flowDirection – Transmit - e.g. one-way speaker call – Receive - e.g. silent monitor service – Transmit&Receive - e.g, barge-in, whisper **Note:** numberOfChanges is not supported
callLinkData	N	Specifies the correlator data associated with the call.
subjectOfCall	N	Specifies the subject/intent of the call.
messageInfo	N	Specifies the contents of message information associated with a call.
privateData	Y	Non-standardized information.

Event Causes

Cause Codes	Retrieved Event	Example Reasons for Cause Code
Alternate	Y	Alternate Call
Key Operation	Y	Bridged Call
Network Signal	N	External Call

Cause Codes	Retrieved Event	Example Reasons for Cause Code
Normal	Y	Any feature, Alternate Call, Reconnect Call, Retrieve Call
Recall	Y	Recall

OpenScape Voice Operational Notes

None

4.6.2.17 Service Initiated

The Service Initiated event indicates that a telephony service has been initiated at a monitored device. This event indicates that either a call may be originated or a feature may be invoked. This event also may indicate that a device is prompting a user.

Event Parameters

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
initiatedConnection	Y	Specifies the connection at which service was initiated.
initiatingDevice	Y	Specifies the initiating device.
localConnectionInfo	Y	Specifies the local connection state of the device associated with the Monitor Cross Reference ID. The following are the states of the devices in the call. When the device is initiating a service of some form: Initiated
cause	Y	The cause parameter specifies the reason for the events
servicesPermitted	Y	Specifies a list of the call control services that can be applied to the local connection.
networkCallingDevice	N	Specifies the original calling device information provided by the network for external incoming calls.
networkCalledDevice	Y	Specifies the original called device information provided by the network for external incoming calls.
associatedCallingDevice	Y	Specifies the Network Interface Device associated with the calling device if the call is an external incoming call. This parameter is mandatory for all external incoming calls and shall not be provided otherwise. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.

Parameter Name	Supported	Comments
associatedCalledDevice	Y	For outgoing external calls, this parameter specifies the Network Interface Device associated with the called device. For incoming external calls, this parameter specifies a device within the switching sub-domain associated with the originally called device. This parameter is mandatory for all external outgoing calls and it is optional for external incoming calls. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.
mediaCallCharacteristics	Y	This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. Supported values: • Voice • Image **Note:** All media associated with a connection are considered initiated by OpenScape Voice.
callCharacteristics	Y	Specifies the characteristics that the switching function reports using the callCharacteristics parameter. Refer to callCharacteristicsSupported parameter description in Section 4.1.1.3, “Get Switching Function Capabilities”.
initiatedConnectionInfo	N	Specifies the connection information associated with the initiated connection.
callLinkData	N	Specifies the correlator data associated with the call.
subjectOfCall	N	Specifies the subject/intent of the call.
messageInfo	N	Specifies the contents of message information associated with a call.
privateData	Y	Non-standardized information. • callbackTarget when prompting is for callback recall.

Event Causes

Cause Code	OpenScape Voice	Example Reasons for Cause Code
Active Participation	Y	Join Call with a participationType of Active (prompting)
Call Back	Y	Call Back Call-Related, Call Back Non-Call-Related

Cause Code	OpenScape Voice	Example Reasons for Cause Code
Call Pickup	Y	Directed Pickup, Group Pickup (prompting)
Conference	Y	Consultation Call with a consultOptions of ConferenceOnly
Consultation	Y	Consultation Call
Join Call	Y	Join Call (prompting)
Make Call	Y	Make Call (prompting)
Make Predictive call	N	Make Predictive Call
New Call	Y	Any Feature
Normal	Y	Any Feature
Reserved	N	Make Predictive Call, incoming call
Silent Participation	Y	Join Call with a participationType of Silent (prompting)
Transfer	Y	Consultation Call with a consultOptions of TransferOnly

Miscellaneous Characteristics

Service initiated shall be provided for prompting is supported.

OpenScape Voice Operational Notes

- Some CSTA services (Make Call and Call Back) may require prompting the user of the targeted device in order to take that device off-hook. In this case a Service Initiated event is generated containing the appropriate cause code (Make Call or Call Back).
- Off-Hook" status of the physical SIP device is not detected by the OpenScape Voice call process. OpenScape Voice sends the service initiated after address translation.
- In the case of a Callback Recall this event provides the callback target device in privateData. The target device number is presented as it appears on the phone.

Note: No name display is provided.

Note: A new ECMA contribution is being considered which adds the "calledDevice" parameter to this event. This change may be added in a future release.

4.6.2.18 Transferred

The Transferred event indicates that an existing call has been transferred to another device and the transferring device has been dropped from the call. The transferring device does not appear in any future events for the call.

Event Parameters

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
primaryOldCall	Y	The switching function provides the local view option. The primaryOldCall specifies the held call at the transferring device, otherwise at other participating devices it is the only call involved in the transfer from the perspective of that device.
secondaryOldCall	Y	The switching function provides the local view option. The secondaryOldCall specifies the active call at the transferring device, otherwise it is not provided.
transferringDevice	Y	Specifies the device that transferred the call.
transferredToDevice *	Y	Specifies the transferred- to device. * See footnote.
transferredConnections	Y	For the transferring device the transferredConnections parameter is a list that contains the new ConnectionIDs and the old ConnectionIDs of the transfer and for externally located devices the associated Network Interface DeviceID. For the other participating devices the old ConnectionID that does not belong to the device is not be provided. The endPoint DeviceID parameter is provided only for external calls with network information. It is the representation of the externally located device. The resultingConnectionInformation: • Supported values for mediaSessionInfo: – Voice – Image • Supported values for flowDirection – Transmit - e.g. one-way speaker call – Receive - e.g. silent monitor service – Transmit&Receive - e.g, barge-in, whisper **Note:** numberOfChanges is not supported

Parameter Name	Supported	Comments
localConnectionInfo	Y	Specifies the local connection state of the device associated with the Monitor Cross Reference ID. The following are the states of the devices in the call. For the transferring device (any Connection IDs associated with the transfer, that is, this event should be used for both single and multi-step transfers.): Null For the other devices associated with the transfer: (unaffected)
cause	Y	The cause parameter specifies the reason for the event.
networkCallingDevice	N	Specifies the original calling device information provided by the network for external incoming calls.
networkCalledDevice	Y	Specifies the original called device information provided by the network for external incoming calls.
associatedCallingDevice	Y	Specifies the Network Interface Device associated with the calling device if the call is an external incoming call. This parameter is mandatory for all external incoming calls and shall not be provided otherwise. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.
associatedCalledDevice	Y	For outgoing external calls, this parameter specifies the Network Interface Device associated with the called device. For incoming external calls, this parameter specifies a device within the switching sub-domain associated with the originally called device. This parameter is mandatory for all external outgoing calls and it is optional for external incoming calls. Parameter not provided if calling and calling device are internal. Not a call involving a gateway. Parameter set to “notKnown” for external incoming or outgoing calls. Calls where a gateway is used.
mediaCallCharacteristics	Y	This specifies the media characteristics to be associated with the call being made. If this parameter is not present then the media class is Voice. Supported values: • Voice • Image **Note:** All media associated with a connection are considered transferred by OpenScape Voice.

Parameter Name	Supported	Comments
callCharacteristics	Y	Specifies the characteristics that the switching function reports using the callCharacteristics parameter. Refer to callCharacteristicsSupported parameter description in Section 4.1.1.3, “Get Switching Function Capabilities”.
callLinkData	N	Specifies the correlator data associated with the call.
servicesPermitted	Y	Specifies a list of the call control services that can be applied to the local connection.
privateData	Y	Non-standardized information.

* If the transferredTo device is a virtual subscriber (Remote Call Forwarding - RCF) then the transferToDevice will be presented as the forwarding target provisioned on the RCF. Refer to MLHG and RCF event flows in CSTA Call Scenarios for additional details.

Event Causes

Cause Codes	Transferred Event	Example Reasons for Cause Code
Network Signal	N	External Call
Normal	Y	Transfer
Path Replacement	Y	SIP and SIP-Q Network Feature
Single Step Transfer	Y	Single Step Transfer
Transfer	Y	Transfer Call, Two Step Transfer

Miscellaneous Characteristics

Local view shall be provided.

OpenScape Voice Operational Notes

None



4.7 Call Associated Features

4.7.1 Services

4.7.1.1 Associate Data

The Associate Data service associates computing function information (such as correlator data, account code, authorization code, call qualifying data, call characteristics, subject of call, language preference, and so on) with a specified call.

The Associate Data service does not affect the state or progress of a call.

Currently the Associate Data service is only supported for Network Interface Devices (NID) when the Name Provider option is active on the associated Endpoint Profile. The deviceInfo parameter shall be used do realize inbound and outbound name provider cases. Refer to the parameter description below.

Service Request Parameters

Parameter Name	Supported	Description
existingCall	Y	Specifies the call with which the information is to be associated. The connectionID must be one previously presented on an NID monitor. **Note:** The deviceID portion of the connectionID must be presented in Device Number (DeN) format. See Section 2.6.1.3, “Device Number (DeN) Format”.
deviceInfo	Y	Specifies the device information (e.g., name) to be associated with the specified connection. The deviceInfo must be presented in SFR format. The name portion of the NM field may be presented in Unicode format. Network Calling Name Provider – The name field will not be accepted if the DN portion does not match the callingDevice DN presented in the Originated event. Network Called Name Provider – The name field will not be accepted if the DN does not match the calledDevice DN presented in the Offer Event.

Parameter Name	Supported	Description
callQualifyingData	Y	Specifies the call qualifying data to associate with an inbound NID call. This parameter is supported by OSV if the existing call connection is associated with an inbound NID call. The parameter format is a hexBinary. To redirect an inbound NID call the format of the string: "altDestination=any dialable number" No other options are currently defined.

Acknowledgement Model

OpenScape Voice V9 supports the multi-step positive acknowledgement model.

CSTA errors are possible for the following:

- existingCall is not a valid NID connection
- deviceInfo DN does not match the callingNetworkDevice DN
- callQualifyingData sent on existingCall that is not on an NID
- callQualifyingData string format/syntax is invalid
- callQualifyingData number is not dialable or otherwise invalid

4.7.1.2 Cancel Telephone Tones

Service Not Supported

The Cancel Telephony Tones service cancels a telephony tone that is being sent on a connection by the Generate Telephony Tones service.

4.7.1.3 Change Connection Information

The Change Connection Information service allows an application to change the connection information associated with a specified connection.

Service Request Parameters

Parameter Name	Supported	Description
connectionToBeChanged	Y	Specifies the connectionID whose connection information is to be changed.
requestedConnectionInfo	Y	Specifies the requested connection information. This information includes the flow direction, number of channels, and the media session information. Supported values for mediaSessionInfo: – Voice – Image Supported values for flowDirection – Transmit - e.g. one-way speaker call – Receive - e.g. silent monitor service – Transmit&Receive - e.g, barge-in, whisper **Note:** numberOfChanges is not supported See usage notes.

Response Parameters

Parameter Name	Supported	Description
actualConnectionInfo	Y	Specifies the actual connection information that is associated with the connection.

OpenScape Voice Operational Notes

- For OpenScape Voice IP Centrex Client, only media session information is supported which contains "voice" or image.
- OpenScape Voice defines Media session information as Character with the following possible strings: "voice" or "image".
- This service may be used to change the media type of an existing call voice of video call. For example:
  - Promoting a voice only call to voice & image (video call)
  - Demoting a voice & image call to voice only
  - Changing voice flow direction for whisper or barge-in features.

- ONS-IO only supports voice media connections.

Attention: One Number Service (ONS) is internal to Unify. Refer to Appendix F, “One Number Service (ONS)” for details.

4.7.1.4 Generate Digits

The Generate Digits service causes a series of digits to be sent on behalf of a connection in a call. The digits may be sent in the form of DTMF. This service also supports optional Parameters to control digit generation. This service is used for generating end-to-end information that is to be sent to a device in a call (that is, not to address/select a device).

This service does not affect the state or progress of a call.

Service Request Parameters

Parameter Name	Supported	Comments
connectionToSendDigits	Y	Connection of the device which is generating the digits for the call.
digitMode	Y	Specifies the signaling format. The complete set of possible values is: DTMF – DTMF Signaling (default)
charactersToSend	Y	Specifies the string of characters to send. Shall consist of the following set: • DTMF digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, #, **Note:** MF digits A, B, C, D are not supported. A comma “,” may be included in the parameter string to indicate a pause between characters. The length of the pause is switching function specific and may be determined using the capabilities exchange services. This parameter type is a character string. The maximum length supported by the switching function is provided using the capabilities exchange services. **Note:** The ampersand (“&”) may be used by the application to cause the immediate release of Media Server resource. In the case were uaCSTA Generate Digit services are provided by the “&” is ignored by the phone.

Response Parameters

None

Miscellaneous Characteristics

Pause tone character of "," supported.

OpenScape Voice Operational Notes

- The charactersToSend string is limited to 32 characters.
- The connection state of both devices in the call must be connected.
- After digit generation has commenced, it cannot be interrupted.
- Successive requests to generate digits is rejected until the active request is complete.
- Only DTMF mode is supported
- Valid characters for DTMF mode are '0' through '9','*','#', and',' (comma). A comma injects a 2 second delay between the signaling of the previous and next digits it separates. Multiple commas can be used to inject longer pauses.
- The duration of DTMF digits and interdigit spacing is determined by the platform media server service.
- When OpenScape Voice provides digit generation using the platform's media service the media sever resource is retained for 3 seconds. During this interval, only ClearConnection and GenerateDigits requests are permitted. The application may request OpenScape Voice to immediately release the media server resources by including an ampersand ("&") character within the charactersToSend string.
- If the target device is a SIP phone that supports uaCSTA, OpenScape Voice may use uaCSTA to generate digits from the device. The CSTA application has no control if uaCSTA is used or not.

4.7.1.5 Generate Telephony Tones

Service Not Supported

The Generate Telephony Tones service causes a telephony tone such as a beep, busy, or ringback to be sent on behalf of a connection in a call. This service also supports optional parameters to control tone generation.

4.7.1.6 Send User Information

Service Not Supported

The Send User Information service is used to send user data information.

4.7.2 Events

4.7.2.1 Call Information

The Call Information event indicates that call associated information has been collected/updated for a call.

Event Parameters

Parameter Name	Supported	Description
monitorCrossRefID	Y	Associates the event to an established monitor.
connection	Y	Indicates the connection of the device responsible for associating the information with the call. **Note:** When presented on a Network Interface device monitor the connection ID is non-standard. In order to ensure that the connectionID presented on the NID monitor the deviceID portion of the connectionIDs is not presented in Device Number (DeN) format, but instead is presented in SFR format. The DN portion of the deviceID shall be the FQDN of the external calling device. In addition a new NM field tag “nid” will include the string representation of the network interface deviceID. Also since the DN is external the “uid” tag is also included in the NM field. For example: X1 calls D2 over NID1 creating connection C1. The Calling deviceID SFR would be presented as: N<X1>;uid=C1;nid=NID1.
device	Y	Indicates the device responsible for associating the information with the call. **Note:** Network Interface devices are presented in Device Number (DeN) format. See Section 2.6.1.3, “Device Number (DeN) Format”.
callingDevice	Y	Indicates the original calling device. OpenScape Voice V9 sends this parameter with the Calling Name after the initial SIP INVITE, to provide updated name information to the application. The Call Information event is only generated if OpenScape Voice V9 has already sent a CSTA Offered, Delivered, Queued or Established event on the called device monitor.
authCode	N	Specifies the authorization code to allow the call.

Parameter Name	Supported	Description
servicesPermitted	Y	Specifies a list of the call control services that can be applied to the local connection.
connectionInfo	Y	Indicates the connection information associated with the connection. Supported values: • Voice • Image
callCharacteristics	Y	Specifies the characteristics that the switching function reports using the callCharacteristics parameter. Refer to callCharacteristicsSupported parameter description inSection 4.1.1.3, “Get Switching Function Capabilities”.
callLinkData	N	Specifies the correlator data associated with the call.
deviceInfo	N	Specifies the device information (for example, name) that has been associated with the connection.
privateData	Y	Non-standardized information.
callQualifyingData	Y	Indicates the call qualifying data to associate with the call. This parameter is supported by OSV if the existing call connection is associated with an inbound NID call and was redirected by the application using the Associated Data service. The parameter format is hexBinary format: • To redirect an inbound NID call the format of the string: "altDestination=any dialable number" No other options are currently defined.

Event Causes

None

OpenScape Voice Operational Notes

The mediaSessionInfo parameter is defined as a string and is encoded for voice, video, and/IM media as follows:

#	Scenario	ConnectionInfo	
		Voice	Image
1	Video has been added to an existing Audio call on the device	✓	+ Added
2	Video has been removed from an existing video call on the device	✓	- Removed
3	Audio has been add to an existing Video only on the device	+ Added	✓
4	Audio has been removed an existing Video Call on the device	- Removed	✓

4.7.2.2 Charging

Event Not Supported

The Charging event indicates that new call charging information (such as a network provided Periodic Pulse Metering (PPM) signal) has been detected for a device involved in a call. This event may occur after the call has been released, to report arrival of additional call charge information.

4.7.2.3 Digits Generated

The Digits Generated event indicates that DTMF digits have been generated at a device.

OpenScape Voice sends this event after Generate Digits service is successfully executed.

Event Parameters

Parameter Name	Supported	Comments
MonitorCrossRefID	Y	Associates the event to an established monitor.
connection	Y	The connection at the device
digitGeneratedList	Y	The sequence of digits generated. For DTMF digits, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, # are supported. This parameter only includes previously unreported digits.

Event Causes

None

OpenScape Voice Operational Note

Refer to notes on Generate Digits for special handling.

4.7.2.4 Telephony Tones Generated

Event Not Supported

The Telephony Tones Generated event indicates that telephony tones have been generated at a device. Common situations that generate this event include; The switching function generates telephony tones (using the Generate Telephony Tones service) for the device of a given connection.

4.7.2.5 Service Completion Failure

The Service Completion Failure event indicates that a previous multi-step computing function initiated service request (as indicated by the switching function in the capabilities exchange services) has failed before that service's successful completion conditions were satisfied.

The Service Completion Failure event SHALL only be reported to the computing function which issues the service request and has a device-type monitor on the device which has or had connection(s) that were used in the particular request (see section 9.2.1.2 Multi-Step Model in ECMA-269).

Event Parameters

The following table has been extracted from ECMA-269 and describes the supported parameters for the Service Completion Failure event. The fields that are stricken through will not be reported by OpenScape Voice V9.

Parameter Name	Type	M/O/C	Description
monitorCrossRefID	MonitorCrossRefID	M	Associates the event to an established monitor.
primaryCall	Structure	M	Specifies information for the connection in the primary call that is associated with the service request. This includes: • deviceID (M) DeviceID - a device involved in the call. • connectionID (M) ConnectionID - a connection at the device. • localConnectionState (M) LocalConnectionState - the local connection state of the connection.
secondaryCall	Structure	O	Specifies information for the connection in the secondary call that is associated with the service request. This includes: • deviceID (M) DeviceID - a device involved in the call. • connectionID (M) ConnectionID - a connection at the device. • localConnectionState (M) LocalConnectionState - the local connection state of the connection. • ~~connectionInformation (O) ConnectionInformation - the connection information for the connection.~~ This parameter may only be present if the service request involved a secondary call.

Parameter Name	Type	M/O/C	Description
~~otherDevicesPrimaryCallList~~	~~List of Structures~~	~~O~~	~~Specifies information for each connection in the primary call. This includes:~~ ~~• deviceID (M) DeviceID – a device involved in the call.~~ ~~• connectionID (M) ConnectionID – a connection at the device.~~ ~~• localConnectionState (O) LocalConnectionState – the local connection state of the connection.~~ ~~• connectionInformation (O) ConnectionInformation – the connection information for the connection.~~ ~~This parameter may only be present if the other connections in the primary call still exist.~~ ~~This parameter may not contain the information associated with the primaryCall parameter.~~
~~otherDevicesSecondaryCallList~~	~~List of Structures~~	~~O~~	~~Specifies information for each of the other connection in the secondary call. This includes:~~ ~~• deviceID (M) DeviceID – a device involved in the call.~~ ~~• connectionID (M) ConnectionID – a connection at the device.~~ ~~• localConnectionState (O) LocalConnectionState – the local connection state of the connection.~~ ~~• connectionInformation (O) ConnectionInformation – connection information for the connectionID.~~ ~~This parameter shall only be present if the service request involved a secondary call and the other connections associated with the secondary call still exist.~~ ~~This parameter shall not contain the information associated with the secondaryCall parameter~~
~~mediaCallCharacteristics~~	~~MediaCallCharacteristics~~	~~O~~	~~Specifies the media class and media characteristics of the call.~~
cause	EventCause	M	Indicates the reason why the service request did not complete.
~~security~~	~~CSTASecurityData~~	~~O~~	~~Specifies timestamp information, message sequence number, and security information.~~

Parameter Name	Type	M/O/C	Description
~~privateData~~	~~CSTAPrivateData~~	~~O~~	~~Specifies non-standardized information with the event.~~

Event Causes

The cause reported in the Service Completion Failure shall indicate the reason why the Seamless Handover has failed, as follows. Note that 'Target OND' below refers to both OND /= ONS and OND == ONS scenarios.

Event Cause	Scenarios
Busy	The Target OND is busy. The Target OND has Call Forwarding Unconditional set on the phone. The Target OND is busy and it has Call Forwarding on Busy set on the phone.
Call Cancelled	The Seamless Handover request has been canceled by the CSTA application via a CSTA Clear Connection request. The original call is cleared before the Seamless Handover is completed. This is the case if either the Starting OND or the Transferred Party release the call before the Seamless Handover is answered at the Target OND.
Call Not Answered	The Target OND does not answer the handover call within 20 seconds. This cause is only sent if the OpenScape Voice V9 received an indication that the Target OND started ringing. The Target OND has Call Forwarding On No Reply set on the phone and the forwarding timer expires.
Do Not Disturb	The Target OND has Do Not Disturb enabled on the phone. **Note:** The server based DND feature is bypassed during a handover.
Incompatible Destination	The target OND is not a valid destination for a Seamless Handover. For instance, the Target OND is a Pilot DN for a MLHG that is not configured as Manual / Application Controlled Mode.
Network Congestion	There are no Call Admission Control (CAC) resources available to connect the Transferred party with the Target OND and rerouting is not available.
Network Not Obtainable	The OpenScape Voice V9 cannot reach the Target OND (e.g. WAN failure, subscriber is unplugged). The following cases fall under this category. • The seamless handover timeout (20 seconds) expires before theOpenScape Voice V9 receives an alerting indication from the Target OND. • The SIP INVITE transaction times out. This can happen before the handover timeout expires depending on how the OpenScape Voice V9 is configured. • The OpenScape Voice V9 receives a SIP 408 Request Timeout response.

Event Cause	Scenarios
Reorder Tone	This shall be the default cause code in case a more specific cause cannot be provided. For instance, this shall be the cause code in case the Target OND rejects the handover call with a SIP 603 Decline response.

Negative Responses for Handover Requests

The following Error Values shall be reported in the negative Single Step Transfer Call responses for both Silent Handover and Seamless Handover requests. Note that these requirements are not applicable for regular Single Step Transfer Call requests (i.e. non-handover).

Error Category	Error Value	Scenarios
Operation Errors	generic	This shall be the default error value in case a more specific error cannot be provided.
	invalidDestination	The Target OND provided in the transferredTo parameter cannot be translated. The Target OND is the same as the Starting OND.
	invalidActiveConnectionIdentifier	The connection specified in the activeCall parameter does not exist.
	invalidDestinationDeviceIdentifier	The device ID in the transferredTo parameter does not match the device ID in the active call. For instance, the application sends a request to handover the call from D1|D2 to D3|D4. The SST request is marked as a Seamless Handover, but there is no OND in the transferredTo parameter.
	privilegeViolationActiveDevice	The device ID in the activeCall parameter is not a subscriber provisioned with the required features: CSTA, ONS Inbound/Outbound and Call Transfer.
	privilegeViolationDestinationDevice	The ONS subscriber is not allowed to call the Target OND due to Toll Restrictions or Station Restrictions.
	requestIncompatibleWithDestinationDevice	The target DN is a valid number, but is not a valid destination for handover (e.g. RCF DN, Vertical Service, etc).

Error Category	Error Value	Scenarios
State Incompatibility Errors	invalidActiveConnectionState	The connection specified in the activeCall parameter is valid, but the requested Handover type (either Silent or Seamless) is not allowed for this connection. Below are some scenarios where the Handover is currently not allowed: • The subscriber is in a station controlled conference (large conference) or in a bridged call. • The subscriber is in a barged-in call. • The subscriber is in a call that has been overridden with Executive Override feature. • The subscriber is either holding the call or is being held. Only applicable for Silent Monitoring requests.
	noActiveCall	The connection specified in the activeCall parameter was valid, but the call is released while processing the Single Step Transfer Call request.
System Resource Availability Errors	resourceBusy	There is a pending request being processed for this call. This could be, for instance, another handover request being processed for the other party. The Target OND is currently being used by a different ONS subscriber in another call.
	destinationDeviceOutOfService	The Target OND is an unregistered subscriber. This is applicable for both OND != ONS and OND == ONS.

4.8 Media Attached Services

Service Not Supported

A computing function making use of both call control services and media services needs to establish sessions with both services, attach calls to the media service, and needs a way of associating the identifiers (e.g., Connection Identifiers, Media Stream Identifiers) used by the two services. This Standard defines a set of services, called media attachment services, that make these tasks significantly easier for the computing function.

4.9 Routing Services

4.9.1 Registration Services

4.9.1.1 Route Register

The Route Register service is used to register the computing function as a routing server for a specific routing device or as a routing server for all routing devices within the switching sub-domain. The computing function may be required to register for routing services before it can receive any route requests for a routing device from the switching function. A computing function may register to be the routing server for more than one routing device.

Note: Application MUST invoke a Route Registration to enable Routing Services. OpenScape Voice will not send Route Request with an active valid Route Registration.

Service Request Parameters

Parameter Name	Supported	Comments
routingDevice	Y	Specifies the routing device for which the computing function requests to be the routing server. This parameter is mandatory if the switching function does not support the option of registering for all routing devices in the switching sub-domain. Otherwise, the parameter is optional and if not present, indicates the registration is to be for all routing devices in the switching sub-domain. **Note:** OpenScape Voice V9 requires that a valid device ID is provided by the application to initiate routing services. If parameter is not presented then request will be rejected.
requestedRouting MediaClass	N	Specifies the media classes of calls that are being requested to be routed. • Voice is default media for routing. • Image (video) media call characteristic may be applied after routing.
security	N	Specifies timestamp information, message sequence number, and security information.
privateData	N	Specifies non-standardized information. No private data is currently defined.

Acknowledgement Model

This service follows the atomic acknowledgement model for this service.

Service Response Parameters

This service follows the atomic acknowledgement model for this service.

Positive Acknowledgement

Parameter Name	Supported	Comments
routeRegisterReqID	Y	Specifies the route register request identifier for this registration. OpenScape Voice V9 supports only one (1) active routeRegisterReqID per routingDevice (subscriber DN), per CSTA connection. The routeRegisterReqID format provided by OpenScape Voice V9 consists of the character “R” + system generated unique ID + routingDevice ServiceID
actualRoutingMediaClass	N	This parameter specifies the actual media classes of calls that are routed by the switching function for routing registration. The actual media classes of calls routed may be the same or a subset of what was requested on the request service. Voice is default media for routing. Image (video) media call characteristic may be applied after routing.
security	N	Specifies timestamp information, message sequence number, and security information.
privateData	N	Specifies non-standardized information. No private data is currently defined.

4.9.1.2 Route Register Abort

This service is used by the switching function to asynchronously cancel an active routing registration. This service invalidates a current routing registration. There is no positive acknowledgement defined for this service.

OpenScape Voice V9 sends this service when a Route Registration is active on a subscriber device and the following occurs:

- CSTA Routing service is removed from the subscriber.
- Subscriber is DN with routing services is removed from the system.

Service Request Parameters

Parameter Name	Supported	Comments
routeRegisterReqID	Y	Specifies the route register request identifier for which the routing registration is to be cancelled. For description details refer to Section 4.9.1.1, “Route Register”.
security	N	Specifies timestamp information, message sequence number, and security information.
privateData	N	Specifies non-standardized information. No private data is currently defined.

Acknowledgement Model

There are no service completion conditions for this service.

OpenScape Voice V9 does not send negative acknowledgement error values defined by ECMA-269.

4.9.1.3 Route Register Cancel

The Route Register Cancel service is used to cancel a previous route registration. This request terminates the routing registration and the computing function receives no further routing requests for that routing registration once it receives the positive acknowledgement to the Route Register Cancel request.

Service Request Parameters

Parameter Name	Supported	Comments
routeRegisterReqID	Y	Specifies the route register request identifier for which the routing registration is to be cancelled. For description details refer to Section 4.9.1.1, “Route Register”.
security	N	Specifies timestamp information, message sequence number, and security information.
privateData	N	Specifies non-standardized information. No private data is currently defined.

Acknowledgement Model

OpenScape Voice V9 supports the atomic acknowledgement model.

Negative acknowledgement error values defined by ECMA-269 are supported.

Service Response Parameters

This service follows the atomic acknowledgement model for this service.

Positive Acknowledgement

Parameter Name	Supported	Comments
security	N	Specifies timestamp information, message sequence number, and security information.
privateData	N	Specifies non-standardized information. No private data is currently defined.

4.9.2 Services

4.9.2.1 Re-Route (with Loop Prevention)

The Re-Route service requests an alternate destination from the one provided by a previous Route Select service and based on previous information provided for the call.

OpenScape Voice V9 ReRoute Loop Prevention:

OpenScape Voice V9 has an internal loop prevention capability to guard against application routing loops. The internal ReRouteLoop count is defaulted to 5 attempts.

The ReRoute loop count is managed via the following RTP parameter:

– hiQ/CSTA/ReRouteLoopCount.

The current loop count is reset if the Route Select service results in the call being delivered to the selectedRoute. The loop count is incremented if the selectedRoute results in an failed call (e.g., busy, out-of-service). If the ReRouteLoop count is exceeded (5 consecutive Route Select failures), then OpenScape Voice V9 invokes Route End service on that routing dialogue.

Service Request Parameters

Parameter Name	Supported	Comments
crossRefIdentifier	Y	Specifies the cross reference identifier associated with the routing dialogue. OpenScape Voice V9 supports only one (1) active crossRefIdentifier per routingDevice (subscriber DN), per CSTA connection. The crossRefIdentifier format provided by OpenScape Voice V9 consists a system generated unique cross-reference ID + “#” + routingDevice ServiceID
routeRegisterReqID	Y	Specifies the route register request identifier associated with the route registration for this routing dialogue. For description details refer to Section 4.9.1.1, “Route Register”.

Parameter Name	Supported	Comments
replyTimeout	N	Specifies the amount of time (in milliseconds) that the switching function will wait for a reply from the computing function, before it proceeds with default routing for the call. If the parameter is not present or the value is 0, the amount of time is switching function specific. CSTA does not expose the configured replyTimeout value. The replyTimeout value is controlled via RTP parameter hiQ/CSTA/DelayAtRouteRequestedState. OpenScape Voice V9 delays call routing (Route Request or ReRoute Request) for approximately 2-4 seconds. Application must provide Route Select, Route End or Route Reject within this timeframe in order to supersede the default routing to the device that initiated the ReRoute request.. If the application fails to respond within the replyTimeout timeframe, OpenScape Voice V9 will invoke a Route End.
correlatorData	N	Specifies the correlator data associated with the call. This parameter is mandatory if there is correlator data associated with the call (and if the parameter is supported).
callLinkageData	N	Specifies the global call data and thread data associated with the call. This parameter is mandatory if the switching function supports the call linkage feature otherwise it shall not be provided.
security	N	Specifies timestamp information, message sequence number, and security information.
privateData	N	Specifies non-standardized information. OpenScape Voice V9 supports ReRouteReason tag with the following values: • busy – routeSelected device is busy • noAnswer – routeSelected device did not answer within the specified Time-To-Answer value provided in the Route Select service. • normalUnspecified – other problems on routeSelected, for example: – device is out-of-service – device is not reachable

Acknowledgement Model

There are no service completion conditions for this service.

OpenScape Voice V9 does not send negative acknowledgement error values defined by ECMA-269.

Refer to replyTimeout parameter description.

Service Response Parameters

This service follows the atomic acknowledgement model for this service.

Positive Acknowledgement

None.

4.9.2.2 Route End

The Route End service ends a routing dialogue. This service is bi-directional (i.e., it may be invoked by the switching function or the computing function). The computing function can use the Route End service when it cannot provide a route for a call. Typically, this can occur if:

- The computing function receives a valid routing request for a call without sufficient call information and it cannot determine a routing destination.
- The computing function has already provided all available destinations for a call and no more alternate destinations are available.
- The computing function does not have access to a database necessary to route the call.

In these cases, the computing function uses the Route End service to inform the switching function that it cannot provide a route for the call in question. The Route End service request will terminate the routing dialogue (routingCrossRefID) for the call. The Route End request does not clear the call. The switching function will continue to process the call using whatever default routing algorithm is available (i.e., in a switching function specific way).

The switching function uses the Route End service when it ends a routing dialogue. Typically, this can occur if:

- The call associated with the routing cross reference identifier has been successfully routed. This may occur when the computing function has sent a Route Select service request and the switching function has successfully routed the call.
- The calling party has abandoned a call associated with the routing cross reference identifier.
- The switching function timeout for a route request response has expired. This may occur if the computing function did not respond to a Route Request or Re-Route Request service within a switching function defined period.
- The switching function has ended routing dialogue due to internal resource (or other) problems. For example, Re-Route Loop Detection, Race conditions (e.g., Answer call before Re-Route; Route Selection.

Service Request Parameters

Parameter Name	Supported	Comments
crossRefIdentifier	Y	Specifies the cross reference identifier associated with the routing dialogue. For description details refer to Section 4.9.2.1, “Re-Route (with Loop Prevention)”.
routeRegisterReqID	Y	Specifies the route register request identifier associated with the route registration for this routing dialogue. For description details refer toSection 4.9.1.1, “Route Register”.
errorValue	Y	Specifies the reason for the route end request. Currently this parameter is defaulted to “generic” in all service requests from OpenScape Voice V9.
correlatorData	N (SF) N (CF)	Specifies the correlator data associated with the call. This parameter is mandatory if there is correlator data associated with the call (and if the parameter is supported).
callLinkageData	N	Specifies the global call data and thread data associated with the call. This parameter is mandatory if the switching function supports the call linkage feature otherwise it shall not be provided.
security	N	Specifies timestamp information, message sequence number, and security information.
privateData	N	Specifies non-standardized information. OpenScape Voice V9 supports routeEndReason tag with the following values: • answered – routed call has been answered. • callCancelled – for example, calling device hangs up, application has no more routing choices. • endCstaDistribution – for example: – call routed to mobile network via PASCODE

Acknowledgement Model

There are no service completion conditions for this service.

OpenScape Voice V9 does not send negative acknowledgement error values defined by ECMA-269.

Service Response Parameters

This service follows the atomic acknowledgement model for this service.

Positive Acknowledgement

None.

4.9.2.3 Route Reject

The Route Reject service request is sent to the switching function during a routing dialogue to indicate that a call should be returned to the originating network (the network from where the call entered the switching sub-domain where the routing request was issued from) for alternate routing.

Service Request Parameters

Parameter Name	Supported	Comments
crossRefIdentifier	Y	Specifies the cross reference identifier associated with the routing dialogue. For description details refer to Section 4.9.2.1, “Re-Route (with Loop Prevention)”.
routeRegisterReqID	Y	Specifies the route register request identifier associated with the route registration for this routing dialogue. For description details refer to Section 4.9.1.1, “Route Register”.
rejectCause	N	Specifies the reason why the call should be returned to the originating network for alternate routing.
correlatorData	N	Specifies the correlator data associated with the call.
security	N	Specifies timestamp information, message sequence number, and security information.
privateData	N	Specifies non-standardized information. None defined.

Acknowledgement Model

There are no service completion conditions for this service.

OpenScape Voice V9 does not send negative acknowledgement error values defined by ECMA-269.

Service Response Parameters

This service follows the atomic acknowledgement model for this service.

Positive Acknowledgement

None.

4.9.2.4 Route Request

The Route Request service requests that the computing function provide a destination for a call. To aid in the selection of a destination, the service request includes the current destination and may include additional information.

OpenScape Voice V9 delays call processing for approximately 2-4 seconds giving the application time to select a route. Refer to the replyTimeout parameter in Section 4.9.2.1, “Re-Route (with Loop Prevention)” for details.

Service Request Parameters

Parameter Name	Supported	Comments
crossRefIdentifier	Y	Specifies the cross reference identifier associated with the routing dialogue. For description details refer to Section 4.9.2.1, “Re-Route (with Loop Prevention)”.
routeRegisterReqID	Y	Specifies the route register request identifier associated with the route registration for this routing dialogue. For description details refer to Section 4.9.1.1, “Route Register”.
currentRoute	Y	Specifies the current destination of the call for which a route is requested. **Note:** OpenScape Voice V9 currently presents the routingDevice as currentRoute
callingDevice	Y	Specifies the originator of the call.
routingDevice	Y	Specifies the device that initiated the Route Request service. **Note:** The routing services are possible regardless of whether a physical SIP device is registered to the subscriber DN. OpenScape Voice V9 supports virtual routing device. However, if the application provides a route selection to the actual routing device where no SIP device is registered, or is currently unreachable, the “Call Forward-Dependable” service should be provisioned to prevent call failure.
routedCall	N	Specifies the ConnectionID of the call. This parameter is mandatory if the route request is call related. If the request is not call-related, then this parameter shall not be provided.

Parameter Name	Supported	Comments
routeSelAlgorithm	N	Specifies the type of routing algorithm requested. The set of possible values are: • ACD • Emergency • Least Cost • Normal • User Defined
associatedCallingDevice	C*	Specifies the Network Interface Device associated with the calling device, if the call is an external incoming call. This parameter is mandatory for all external incoming calls. Otherwise, it is not provided. * OpenScape Voice V9 presents this parameter if the calling device is across a network interface (gateway). In this case the parameter value = “notKnown”.
associatedCalledDevice	N	For outgoing external calls, this parameter specifies the Network Interface Device associated with the originally called device. For incoming external calls, this parameter specifies a device within the switching sub-domain associated with the originally called device. This parameter is mandatory for all external outgoing calls and, is optional for external incoming calls.
priority	N	Specifies the call priority. This may affect the selection of alternative routes.
replyTimeout	N	Specifies the amount of time (in milliseconds) that the switching function will wait for a reply from the computing function, before it proceeds with default routing for the call. If the parameter is not present or the value is 0, the amount of time is switching function specific. For description details refer to Section 4.9.2.1, “Re-Route (with Loop Prevention)”
correlatorData	N	Specifies the correlator data associated with the call. This parameter is mandatory if there is correlator data associated with the call (and if the parameter is supported).
mediaCallCharacteristics	N	This specifies the media class (voice, digital data, etc.) and characteristics of the call. If this parameter is not present (and the parameter is supported) then the call by default is a voice call. OpenScape Voice V9 does not present this information for routing services. However, media class of calling party is presented in the call control events.
callCharacteristics	N	Specifies the high level characteristics (ACD call, Priority call, etc.) associated with the call.

Parameter Name	Supported	Comments
routedCallInfo	N	Specifies the connection information associated with the routedCall connection. If this parameter is not present, then the connection information is switching function specific.
callLinkageData	N	Specifies the global call data and thread data associated with the call. This parameter is mandatory if the switching function supports the call linkage feature. Otherwise, it shall not be provided.
subjectOfCall	N	Specifies the subject/intent associated with the call. This for example, could represent the subject header of an Email (text call) or intent for a voice call.
messageInfo	N	Specifies the contents of message information associated with a call. The message information consists of one of more items.
languagePreferences	N	Specifies the language preferences associated with the call.
deviceHistory	N	Specifies the list of devices which were previously associated with the call (e.g. redirecting, transferring, clearing devices).
security	N	Specifies timestamp information, message sequence number, and security information.
privateData	N	Specifies non-standardized information. None defined.

Acknowledgement Model

There are no service completion conditions for this service.

OpenScape Voice V9 does not send negative acknowledgement error values defined by ECMA-269.

Refer to the replyTimeout parameter in this section under Service Request Parameters.

Service Response Parameters

This service follows the atomic acknowledgement model for this service.

Positive Acknowledgement

None.

4.9.2.5 Route Select

The Route Select service is used by the computing function to provide the destination requested by a previous Route Request or Re-Route service.

Service Request Parameters

Parameter Name	Supported	Comments
crossRefIdentifier	Y	Specifies the cross reference identifier associated with the routing dialogue. For description details refer to Section 4.9.2.1, "Re-Route (with Loop Prevention)".
routeRegisterReqID	Y	Specifies the route register request identifier associated with the route registration for this routing dialogue. For description details refer to Section 4.9.1.1, "Route Register".
routeSelected	Y	Specifies the primary selected destination of the call for which a route was requested. --- **Note:** If the routeSelected device is the same as one in the previous Route Select on this routing dialogue, then OpenScape Voice V9 invokes a new Re-Route Request. The replyTimeout is not affected and the application must respond within the Route Select remaining time. ---
alternateRoutes	N	Specifies the list of alternate destinations (in priority order) which are to be used sequentially for routing the call if the primary selected destination initially (i.e., the routeSelected parameter) or a previous entry in the list is not valid or available.
remainRetries	N	Specifies either the number of alternative routes remaining or the reason why the computing function is not providing it.
routeUsedReq	N	Specifies whether the switching function should issue the Route Used service when it has accepted a route.
correlatorData	N	Specifies the correlator data associated with the call. This parameter is mandatory if there is correlator data associated with the call (and if the parameter is supported).
security	N	Specifies timestamp information, message sequence number, and security information.

Parameter Name	Supported	Comments
privateData	N	Specifies non-standardized information. The following tags and values are: • Time-to-Answer: – If value is 0 then the call is sent to the selected device. OpenScape Voice V9 will not provide answer supervision for re-routing. – If value is >0 seconds then OpenScape Voice V9 will invoke ReRoute when timer expires. • Request_Supression_Indicator: – True; OSV will continue current routing dialogue and will NOT start a new routing dialog on selected device. – False; OSV will invoke a route end on the current routing dialogue and start a new routing dialogue on the selected device. **Note:** If privateData is not provided then OpenScape Voice V9 shall treat the Route Select options as Time-to-Answer = 0 and RSI=True.

Acknowledgement Model

There are no service completion conditions for this service.

OpenScape Voice V9 does not send negative acknowledgement error values defined by ECMA-269.

Service Response Parameters

This service follows the atomic acknowledgement model for this service.

Positive Acknowledgement

None.

4.9.2.6 Route Used (not supported)

The Route Used service provides the actual destination for a call that has been routed using the Route Select service with its optional parameter that requests the route that was used.

4.10 Physical Device Features

4.10.1 Services

This section describes the supported CSTA Physical Device features. Physical device features are only supported by uaCSTA-enabled SIP phones such as OpenStage/

OpenScape Voice does **not support** the following physical device services:

- Button Press
- Get Auditory Apparatus Information
- Get Button Information
- Get Display
- Get Hookswitch Status
- Get Lamp Information
- Get Lamp Mode
- Get Microphone Gain
- Get Ringer Status
- Get Speaker Mute
- Set Button Information
- Set Display
- Set Hookswitch Status
- Set Lamp Mode
- Set Message Waiting Indicator
- Set Microphone Gain
- Set Ringer Status
- Set Speaker Mute

4.10.1.1 Get Message Waiting Indicator

The Get Message Waiting Indicator (MWI) service provides the message waiting feature status at a specified device. The message waiting feature is typically used to notify a user (typically using a dedicated lamp on a phone device) when messages are available.

Service Request Parameters

Parameter Name	Supported	Comments
device	Y	Specifies the device's physical element.

Response Parameters

Parameter Name	Supported	Comments
messageWaitingOn	Y	Specifies the value (True / False) of the requested feature. Supported values: • FALSE – Message waiting is off • TRUE – Message waiting is on
deviceForMessage	N	Specifies the device where the message is waiting.

OpenScape Voice Operational Notes

- OpenScape Voice does not currently support the Set Message Wait Service.
- The Message Waiting event is flowed by OpenScape Voice when a subscriber with voice messaging service is notified by the service of a change in MWI state.

4.10.1.2 Get Microphone Mute

The Get Microphone Mute Service provides the Mute status of a specified microphone at a specified device.

Service Request Parameters

Parameter Name	Supported	Comments
Device	Y	Specifies the device's physical element. The device provided here must be a SIP endpoint
auditoryApparatus	Y	Specifies the auditory apparatus to query. If not provided, then information is obtained on all apparatuses associated with the device.

Response Parameters

Parameter Name	Supported	Comments
microphoneMuteList	Y	Specifies information about the specified auditory apparatus or about all auditory apparatuses associated with the device if no auditoryApparatus was provided in the request. Supported options: • auditoryApparatus • microphoneMuteOn – TRUE – microphone muted – FALSE – microphone activated

OpenScape Voice Operational Notes

- This service is only supported for SIP phones that register with SIMPLE uaCSTA capability.
- The auditoryApparatus is optional. If provided the value should be set to "1". Either option results in a list of status for all microphones being returned. Other values results in an error being returned.

4.10.1.3 Set Message Waiting Service

The Set Message Waiting Indicator service allows a computing function to control the status of the message waiting feature at a specified device. The message waiting feature is typically used to notify a user (typically via a dedicated light on a phone device) when messages are available.

Service Request Parameters

Parameter Name	Supported	Comments
device	Y	Specifies the device's physical element.
messageWaitingOn	Y	Specifies the setting of the message waiting feature. The set of possible values are: • OFF - Message waiting off. • ON - Message waiting on.
deviceForMessage	N	Specifies the device where the message is waiting. OpenScape Voice V9 Call Forward – Voice Mail service must be provisioned and active.
security	N	Specifies timestamp information, message sequence number, and security information.
privateData	Y	Specifies non-standardized information. **privateData tag: numberOfWaitingMessages:** The value of this tag may be presented in the format defined in RFC 3842 for voice messages: • "Voice-Messages: new messages / old messages (urgent new /urgent old)" Examples: • If the application wants to report 1 new, 3 old, 0 urgent new and 1 urgent old the private data format should be presented as follows: – "Voice-Messages: 1/3 (0/1)" • If urgent message data is not provided by the application then the format should be presented as follows: – "Voice-Messages: 1/3" • If only new messages are presented then OSV supports the following format: – "Voice-Messages: 1" • f this parameter is omitted then no message count is sent to the phone for display.

Acknowledgement Model

OpenScape Voice V9 supports the atomic positive acknowledgement model.

Negative acknowledgement error values defined by ECMA-269 are supported.

Service Response Parameters

This service follows the atomic acknowledgement model for this service.

Positive Acknowledgement

Parameter Name	Supported	Comments
security	N	Specifies timestamp information, message sequence number, and security information.
privateData	N	Specifies non-standardized information. None defined.

OpenScape Voice Operational Notes

- OpenScape Voice V9 sends MWI event in response the Set MWI request.
- OpenScape Voice V9 shall maintain the last MWI state provided by the application internally and on the registered SIP phone. MWI is maintained across OpenScape Voice V9 restarts. Applications may use Get MWI to verify the current status.
- OpenScape Voice V9 retains the MWI status over switch and phone restarts. The number of messages is not retained.
- It is the application's responsibility to maintain the MWI synchronize with the non-OpenScape Voice V9, voice messaging service.
- Unify SIP support MWI buttons that are capable of accessing external VM systems.

4.10.1.4 Get Speaker Volume

The Get Speaker Volume Service provides the volume of a specified speaker at a specified device.

Service Request Parameters

Parameter Name	Supported	Comments
Device	Y	Specifies the device's physical element.
auditoryApparatus	Y	Specifies the auditory apparatus to query. If not provided, then information is obtained on all apparatuses associated with the device.

Response Parameters

Parameter Name	Supported	Comments
speakerVolumeList	Y	Specifies information about the specified auditory apparatus or about all auditory apparatuses associated with the device if no auditoryApparatus was provided in the request. Supported options: • auditoryApparatus • speakerVolAbs – A value of 0-100 where 0 indicates no volume and 100 indicates maximum volume

OpenScape Voice Operational Notes

- This service is only supported for SIP phones that register with SIMPLE uaCSTA capability.
- On the request, auditoryApparatus can be either "1" or can be absent. Both result in the status of the current active speaker being returned.

4.10.1.5 Set Microphone Mute

The Set Microphone Mute Service sets or clears the Mute status of a specified microphone at a specified device. When the Mute feature is activated at this device, no audio is transmitted using the microphone. This feature is used when it is desired to prevent the other parties in a call from hearing a conversation through the device.

Service Request Parameters

Parameter Name	Supported	Comments
device	Y	Specifies the device's physical element.
auditoryApparatus	Y	Specifies the auditory apparatus to query. Must be "1" indicating all microphones.
microphoneMuteList	Y	Specifies the microphone mute setting of a particular microphone. • microphoneMuteOn – TRUE – mute microphone – FALSE – activate microphone

Response Parameters

None

OpenScape Voice Operational Notes

- This service is only supported for SIP phones that register with SIMPLE uaCSTA capability.
- On the request, auditoryApparatus must be "1" indicating all microphones.

4.10.1.6 Set Speaker Volume

The Set Speaker Volume Service sets the volume of the specified speaker at a specified device.

While a device's microphone is muted, no audio information is transmitted over the microphone. This is used when it is desired to prevent the other device(s) in a call from hearing a conversation.

Service Request Parameters

Parameter Name	Supported	Comments
Device	Y	Specifies the device's physical element.
auditoryApparatus	Y	Specifies the auditory apparatus to query. Must be "1" indicating the current active speaker.
speakerVolume	Y	Specifies the speaker volume as an absolute value or that the volume should be incremented or decremented by a switch specified increment. • speakerVol – 0-100 OR • speakerVol – increment – decrement

Response Parameters

None

OpenScape Voice Operational Notes

- This service is only supported for SIP phones that register with SIMPLE uaCSTA capability.
- On the request, auditoryApparatus must be "1" indicating the current active speaker. If speakerVolInc is used, the increment or decrement level is determined by the phone.

4.10.2 Events

This section describes the supported CSTA Physical Device Events.

OpenScape Voice does **not support** the following physical device feature events:

- Button Information
- Button Press
- Display Updated
- Hookswitch
- Lamp Mode
- Microphone Gain
- Microphone Mute
- Ringer Status
- Speaker Mute
- Speaker Volume

4.10.2.1 Message Waiting

The Message Waiting event indicates that the message waiting status has been changed for a device.

Event Parameters

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
targetDevice	Y	Specifies the device where the message waiting feature has changed.
messageWaitingOn	Y	Specifies the setting of the message waiting feature. Supported values: • FALSE - Message waiting is off. • TRUE – Message waiting is on.

Event Causes

None

OpenScape Voice Operational Note

The Message Waiting event is sent when a subscriber provisioned for voice messaging and the messaging service updates the MWI status. OpenScape Voice supports SIP Blind NOTIFY for MWI sent from SIP voice messaging servers' service to the subscriber DN. Refer to the *OpenScape Voice Security Checklist*.

4.11 Logical Device Feature Services

4.11.1 Services

This section describes the supported CSTA Logical Device Services.

OpenScape Voice does not support the following physical device services:

- Call Back Message Non-Call-Related
- Cancel Call Back Message
- Get Auto Answer
- Get Auto Work Mode
- Get CallBack (ECMA 269 edition 7)
- Get Caller ID Status
- Get Last Number Dialed
- Get Routing Mode
- Set Auto Answer
- Set Auto Work Mode
- Set Caller ID Status
- Set Routing Mode

4.11.1.1 Call Back Non-Call Related

The Call Back Non-Call-Related service allows a computing function to request that the switching function originate a call back call between two devices.

As an example, the service might be used when a device is busy so that a call between an originating device and a target device can be attempted when a device becomes free.

Service Request Parameters

Parameter Name	Supported	Comments
originatingDevice	Y	Specifies the originating device for the call back call.
targetDevice	Y	Specifies the target device for the call back call. Only last called deviceID or the string “CCB” is supported. (See OpenScape Voice V9 Operational Notes).

Response Parameters

None

OpenScape Voice Operational Notes

- The Call Back Non-Call Related service only supports a targetDevice equal to the last called number. A service request where the targetDevice does not equal (after translation) the last saved dialed number or a previously set Call Back, will be rejected with:
  - CSTA errorvalue= requestIncompatibleWithDevice.
- A hardcoded tragetDevice “CCB” may be used by applications to simplify the call back non-call related service invocation.

4.11.1.2 Cancel Call Back

The Cancel Call Back service allows the computing function to cancel a previous (or all) Call Back feature at a device. Note that this service cancels call backs that were created with either call related or non-call related Call Back features.

Service Request Parameters

Parameter Name	Supported	Comments
originatingDevice	Y	DeviceID of the device who initiated the original Call Back service (call or non-call related)
targetDevice	Y	The DeviceID of the target of the original Call Back service. If the switching function supports clearing of all Call Back features (as indicated by the capability exchange services) and a null format DeviceID (a DeviceID with 0 characters) is provided, all of the Call Back features at the originatingDevice are cancelled. OpenScape Voice supports both methods of call back cancellation.

Response Parameters

None

OpenScape Voice Operational Notes

- OpenScape Voice supports two methods of canceling callback originated by the originatingDevice:
  - Cancel specific call back when the targetDevice = valid deviceID in the callback list
  - All call backs when the targetDevice has 0 characters.
- In the case of cancel all, if a callback is in the process of recalling or being established, these are not cancelled.

4.11.1.3 Get Agent State

The Get Agent State service provides the agent state at a specified device.

Note: The agent may only be associated with one device.

Service Request Parameters

Parameter Name	Supported	Comments
device	Y	Specifies the DeviceID of the device on which the agent state is being queried.

Response Parameters

Parameter Name	Support	Comments		
agentStateList	Y	This parameter specifies a list of agent identifies and / or their corresponding agent state for a given device. Supported parameters:		
		Parameter	**Support**	**Values**
		agentID	Y	DeviceID
		loggedOnState	Y	• TRUE = logged on • FALSE = not logged **NOTE:** For agents (multi-line hunt group members with the attendant classmark) in hunt groups managed by OpenScape Voice this is always set to TRUE.
		acdGroup	Y	The acdGroup is provided if the agent is busy with a hunt group call.
		agentState	Y	Supported States: • Ready • Not Ready • Busy • Null • Working After Call
		pendingAgentState	Y	Indicates pending state if agentState is Busy or Working After Call. This component shall be provided if the switching function is delaying the transition to the pendingAgentState until the agent is no longer Busy or Working After Call, otherwise the parameter is optional. Supported states: • Working After Call • Not Ready • Ready • Null.

Parameter Name	Support	Comments
privateData	Y	extendedMakeBusyStatusList: • extendedMakeBusyStatus - repeated for each hunt group where agentID is a member – groupDn – deviceID; FQN of hunt group – groupName – string; provisioned hunt group name – makeBusy – Boolean; true (Not Ready) or, false (Ready)

OpenScape Voice Operational Notes

- Agent states Not Ready, Ready and Working after Call depends on proper provisioning of MHLG. Refer to Section 3.2.3, “MLHG Pilot (Group Device / ACD) and Agent BGLs” for details.
- Agent Busy state applies any agent direct call (inbound or outbound) or call distributed to an agent from the MLHG.
- An administrable CSTA application interface option named “Enabled Extended Agent States” must be enabled to extendedMakeBusyStatusList privateData presentation.
- See Part II, Appendix A, “CSTA Call Scenarios”, for usage notes.

4.11.1.4 Set Agent State

The Set Agent State service requests a new agent state at a specified device.

In the case where an agent is involved with a call, the transition to the requested state may or may not occur until the current connection transitions to the null state.

Service Request Parameters

Parameter Name	Supported	Comments
device	Y	Specifies the DeviceID for the agent for which the state is to be changed.
requestedAgentState	Y	Specifies the requested agent state. Currently supported values are • Ready • NotReady.
agentID	N	Specifies the agent identifier.
password	N	Specifies the agent password. This parameter can only be provided when the requestedAgentState is loggedOn or loggedOff.
group	N	Specifies the agent ACD group that the agent is logging in or out of.
privateData	Y	• huntGroup (deviceID) - FQN of the Hunt Group where the requestAgentState is to be applied.

Response Parameters

Parameter Name	Supported	Comments
pendingAgentState	Y	Indicates the agent state that the agent may transition to after the agent state is no longer Busy or Working After Call. Supported values: • workingAfterCall • notReady • ready

Miscellaneous Characteristics

Description	Supported
Switching function allows a group (ACD group) device in the service request (that is service applies to all agent in ACD group).	Y
Switching function allows an ACD device in the service request (that is service applies to all agents associated with the ACD device)	Y
Switching function delays transition to the requestedAgentState if it is Busy (that is supports the pending agent state)	Y

OpenScape Voice Operational Notes

- Agent states Busy and Working After Call depend on proper provisioning of MHLG. Refer to Section 3.2.3, “MLHG Pilot (Group Device / ACD) and Agent BGLs” for details.
- Agent Busy state applies any agent direct call (inbound or outbound) or call distributed to an agent from the MLHG.
- An administrable CSTA application interface option named “Enabled Extended Agent States” must be enabled to huntGroup privateData support.
- See Part II, Appendix A, “CSTA Call Scenarios”, for usage notes.

CSTA Error Responses

Item	Condition	CSTA Error
1	The huntGroup parameter is presented by the application and the "Enable Extended Agent States" is disabled	agentGroupNotSupported
2	The huntGroup parameter is presented but is not a valid Hunt Group Pilot	invalidAgentGroup
3	The subscriber is not provisioned to set Agent State (existing functionality)	privilegeViolationSpecifiedDevice
4	The requested state is already set for the specified huntGroup (existing functionality)	featureAlreadySet

4.11.1.5 Get Do Not Disturb

The Get Do Not Disturb service provides the do not disturb feature status at a specified device. The do not disturb feature is used to prevent incoming calls at a device.

The privateData element named DNDResume has been introduced in the service response. The DNDResume parameter identifies the remaining number of minutes the DND feature will stay active.

When the DNDResume time expires the OSV DND feature is turned off for the specified device.

The presentation of the DNDResume parameter is controlled by an OSV provisioned CSTA Interface Option.

Service Request Parameters

Parameter Name	Supported	Comments
device	Y	Specifies the DeviceID of the device on which the do not disturb feature is being queried.

Response Parameters

Parameter Name	Supported	Comments
doNotDisturbOn	Y	Specifies the value of the requested feature. Supported values: • TRUE – Do Not Disturb is active • FALSE – Do Not Disturb is inactive.
privateData	Y	The DNDResume parameter is supported to specific time duration for the DND feature to remain active. This parameter is an integer type value. The rules for presentation are as follows: • "If the doNotDisturbOn=FALSE then the DNDResume parameter is presented as null parameter. • "If the doNotDisturbOn=TRUE then the DNDResume contains a value of 0-1440 minutes, which represents the remaining snooze time.

XML Example - CSTA over TCP

```
<?xml version="1.0" encoding="utf-8"?>
<GetDoNotDisturb>
  <device>15619231439</device>
</GetDoNotDisturb>

<?xml version="1.0" encoding="UTF-8"?>
<GetDoNotDisturbResponse xmlns="http://www.ecma-
international.org/standards/ecma-323/csta/ed4">
```

```
  <doNotDisturbOn>true</doNotDisturbOn>
   <extensions>
    <privateData>
      <private xmlns:scx="http://www.siemens.com/schema/csta">
        <scx:DNDResume>60</scx:DNDResume>
      </private>
    </privateData>
  </extensions>
</GetDoNotDisturbResponse>
```

XML Example - CSTA over SIP

Note: NOTE: The short tag name for DNDResume privateData is zDRm.

```
<?xml version="1.0" encoding="utf-8"?>
<GDND>
  <dvc>15619231439</dvc>
</GDND>

<?xml version="1.0" encoding="UTF-8"?>
<GDNDe xmlns="http://www.ecma-international.org/standards/ecma-
323/csta/ed4">
  <dNDO>true</dNDO>
   <extn>
    <pDa>
      <prvt xmlns:scx="http://www.siemens.com/schema/csta">
        <scx:zDRm>60</scx:zDRm>
      </prvt>
    </pDa>
  </extn>
</GDNDe>
```

OpenScape Voice Operational Notes

- If the device is an internal MLHG (Hunt Group) the Get DND service may be used to provide status of the Night Service overflow for the group.

To synchronize OpenScape Voice centralized Call Forwarding with an OpenStage the CSTA device category must be set to “CSTA over SIP”.

4.11.1.6 Set Do Not Disturb

The Set Do Not Disturb service allows the computing function to control the do not disturb feature at a specified device. The do not disturb feature is typically used to prevent a specified device from being alerted.

The privateData named DNDResume has been introduced in the service request. The DNDResume sets the number of minutes the DND feature will stay active.

If a prior DND-snooze is active a subsequent will keep DND active and apply the new specified snooze duration from the "DNDResume" parameter.

When the DNDResume time expires the OSV DND feature is turned off for the specified device.

The acceptance of the resume parameter in this service is controlled by an OSV provisioned CSTA Interface Option.

Service Request Parameters

Parameter Name	Supported	Comments
device	Y	Specifies the DeviceID of the device on which the do not disturb feature is to be set
doNotDisturbOn	Y	Specifies the value of the requested feature. Supported values: • TRUE – Do Not Disturb is active • FALSE – Do Not Disturb is inactive.
privateData	Y	The DNDResume parameter is supported to specific time duration for the DND feature to remain active. OSV ignores by OSV if doNotDisturbOn=FALSE it is only applied during DND feature activation. The range for the resume value is 0-1440 indicating the number minutes that the Do Not Disturb feature shall remain active. • "If the DNDResume parameter is omitted, null or set to zero (0) no timer will be started and the user must manually cancel the Do Not Disturb feature. • "If the DNDResume value is >1440 minutes CSTA SM shall reject the request with a CSTA Private Data Information Error = cSTAPrivateDataInfoError.

Response Parameters

None

XML Example - CSTA over TCP

```
<?xml version="1.0" encoding="utf-8"?>
<SetDoNotDisturb>
  <device>15619231439</device>
  <doNotDisturbOn>true</doNotDisturbOn>
```

```
   <extensions>
    <privateData>
      <private xmlns:scx="http://www.siemens.com/schema/csta">
        <scx:DNDResume>60</scx:DNDResume>
      </private>
    </privateData>
  </extensions>
</SetDoNotDisturb>


<SetDoNotDisturbResponse xmlns="http://www.ecma-
international.org/standards/ecma-323/csta/ed4"/>
```

XML Example - CSTA over SIP w/short tags

Note: The short tag name for DNDResume privateData is zDRm.

```
<?xml version="1.0" encoding="utf-8"?>
<sDND>
  <dvc>15619231439</dvc>
  <dNDO>true</dNDO>
   <extn>
    <pDa>
      <prvt xmlns:scx="http://www.siemens.com/schema/csta">
        <scx:zDRm>60</scx:zDRm>
      </prvt>
    </pDa>
  </extn>
</sDND>


<SDNDe xmlns="http://www.ecma-international.org/standards/ecma-
323/csta/ed4" />
```

OpenScape Voice Operational Notes

- Subscriber DN must be provisioned with DND service.
- ~~If the device is an internal MLHG (Hunt Group) the Set DND service may be used to activate or deactivate Night Service overflow for the group.~~
- To synchronize OpenScape Voice centralized Do Not Disturb status with an OpenStage device, the CSTA device category must be set to “CSTA over SIP”.

4.11.1.7 Get Forwarding

The Get Forwarding service provides the forwarding feature indication for a specified device. The status returned may consist of one or more forwarding types that are active at the specified device based on user defined conditions. The forwarding feature is used to redirect calls that arrive at a specified device to an alternate destination.

Service Request Parameters

Parameter Name	Supported	Comments
device	Y	Specifies the device on which to query.

Response Parameters

<table>
<tr><th>Parameter Name</th><th>Supported</th><th colspan="3">Comments</th></tr>
<tr><td rowspan="8">forwardList</td><td rowspan="8">Y</td><td colspan="3">This list contains one structure per forwardingType/ forwardDN combination. Supported values:</td></tr>
<tr><td>Parameter</td><td>Support</td><td>Values</td></tr>
<tr><td>forwardingType</td><td>Y</td><td>• forwardImmediate
• fowardBusy
• fowardNoAns
• forwardBusyInt
• forwardBusyExt
• forwardNoAnsInt
• forwardNoAnsExt
• forwardImmInt
• forwardImmExt</td></tr>
<tr><td>forward Status</td><td>Y</td><td>• TRUE = forwarding is active
• FALSE = forwarding is inactive</td></tr>
<tr><td>forwardTo</td><td>Y</td><td>DeviceID</td></tr>
<tr><td>fowardDefault</td><td>N</td><td>n/a</td></tr>
<tr><td>ringCount</td><td>N</td><td>n/a</td></tr>
<tr><td>ringDuration</td><td>N</td><td>n/a</td></tr>
</table>

Parameter Name	Supported	Comments

privateData	Y	The following is presented for standard call forwarding types:
		• setForwardingPermitted
		  • forwardingTypeImmediate; boolean
		  • forwardingTypeBusy; boolean
		  • forwardingTypeNoAnswer; boolean
		  • forwardingTypeImmediateInt; boolean
		  • forwardingTypeImmediateExt; boolean
		  • forwardingTypeBusyInt; boolean
		  • forwardingTypeBusyExt; boolean
		  • forwardingTypeNoAnswerInt; boolean
		  • forwardingTypeNoAnswerExt; boolean
		Supports all 9 forwarding types as in the forwardList. This Indicates whether the specific forwarding type is configurable by endpoints.
		
		The following is presented for ONS subscribers only:
		• staticOndActive: Feature active, True or False.
		• staticOndDN: current dialable number, i.e., target OND.

CSTA Error Responses

CSTA Error Codes		Condition
CSTA Error	**CSTA Error Values**	
operation	invalidDeviceID	The identified device in the CSTA request is invalid
	serviceNotSupported	The identified device in the CSTA request is not provisioned for this feature

OpenScape Voice Operational Notes

- CSTA exposes OpenScape Voice Call Forwarding status for Immediate, No Answer and Busy variants only.
- To synchronize OpenScape Voice centralized Call Forwarding with an OpenStage device, the CSTA device category must be set to “CSTA over SIP”.

4.11.1.8 Set Forwarding

The Set Forwarding service allows the computing function to control the forwarding feature at a specified device based on user defined conditions. The forwarding feature is used to redirect calls that arrive at a specified device to an alternate destination. This service allows only one user-specified setting (forwarding type/forward-destination combination) to be changed per service invocation.

Service Request Parameters

Parameter Name	Supported	Comments
device	Y	Specifies the device on which to set the feature.
forwardingType	Y	Specifies the type of forwarding. Supported values: • forwardImmediate • fowardBusy • fowardNoAns • forwardBusyInt • forwardBusyExt • forwardNoAnsInt • forwardNoAnsExt • forwardImmInt • forwardImmExt The “internal” and “external” types are lower precedence types that refer to the type of call origination (for example an external call) that will be forwarded if it matches a forwarding type (for example forwardImmExt) enabled at the device provided that the higher precedence type (e.g.forwardImmediate) had not already forwarded the call. **Note:** This parameter is executed when setting Call Forwarding Static OND.
activateForward	Y	Indicates the status of the forwarding type. Supported values: • True: Activate forwarding • False: Deactivate forwarding **Note:** This parameter is ignored when setting Call Forwarding Static OND is set.
forwardDN	Y	Specifies the device to which new calls are forwarded. This parameter shall be provided for variable forwarding settings when activateForward is set to TRUE or is optional if set to FALSE.

Parameter Name	Supported	Comments
ringCount	N	Specifies the number of times a device should ring prior to forwarding no answer.
ringDuration	N	Specifies the time (in seconds) that the device should ring prior to forwarding no answer. The ringDuration is not be provided if ringCount is provided.
privateData	Y	Supported for Call Forward - Static OND: • staticOndActive: Used to activate Call Forward - Static OND; True or False. • staticOndDN: blank or dialable number, used to set the target OND for inbound calls.

Response Parameters

None

CSTA Error Responses

CSTA Error Codes		Condition
CSTA Error	**CSTA Error Values**	
operation	invalidDeviceID	The identified device in the CSTA request is invalid
	privilegeViolation	The identified device in the CSTA request is not provisioned for this feature
	invalidForwardingType	Specified forwarding Type not supported
	invalidForwardingFlag	Specified forwarding Flag not supported
	invalidForwardingDestination	Specified forwardTo deviceID is invalid
	invalidRingDuration	RingDuration invalid

OpenScape Voice Operational Notes

- CSTA exposes OpenScape Voice Call Forwarding activation / deactivation for Immediate, No Answer and Busy variants only.
- To synchronize OpenScape Voice centralized Call Forwarding with an OpenStage device, the CSTA device category must be set to “CSTA over SIP”.

4.11.2 Events

This section describes the supported CSTA Logical Device Events.

OpenScape Voice does **not support** the following physical device feature events:

- Agent Logged On
- Agent Logged Off
- Auto Answer
- Auto Work Mode
- Call Back Message
- Caller ID Status
- Routing Mode

4.11.2.1 Agent Busy

The Agent Busy event indicates that an agent has entered the Busy state. In this state an agent is involved with an existing call at a device, even if that call is on hold at the device.

Event Parameters

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
agentDevice	Y	Indicates the device ID at which the agent entered the Agent Busy state.
pendingAgentState	Y	Indicates the agent state that the agent transitions to after the agent state is no longer Busy. Supported states: • Working After Call • Not Ready • Ready • Null.
agentID	N	Indicates the agent identifier.
acdGroup	Y	Indicates the ACD group, if call was distributed from MLHG.
cause	Y	Indicates a reason for the event.
privateData	Y	extendedMakeBusyStatusList: • extendedMakeBusyStatus - repeated for each hunt group where agentID is a member – groupDn – deviceID; FQN of hunt group – groupName – string; provisioned hunt group name – makeBusy – Boolean; true (Not Ready) or, false (Ready)

Event Causes

Cause code = Normal.

OpenScape Voice Operational Note

- Agent Busy state applies any agent direct call (inbound or outbound) or call distributed to an agent from the MLHG
- An administrable CSTA application interface option named “Enabled Extended Agent States” must be enabled extendedMakeBusyStatusList privateData presentation.
- See Part II, Appendix A, “CSTA Call Scenarios”, for more information.

4.11.2.2 Agent Not Ready

The Agent Not Ready event indicates that an agent has entered the Agent Not Ready state. While in this state an agent may receive calls that are not distributed from an associated group device.

Event Parameters

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
agentDevice	Y	The DeviceID specifies the directory number of the agent device that is Not Ready.
agentID	N	Indicates the agent identifier.
acdGroup	N	Indicates the ACD group. Agent is Not Ready for all groups where it is a member.
cause	Y	Indicates a reason for the event. Always Normal.
privateData	Y	extendedMakeBusyStatusList: • extendedMakeBusyStatus - repeated for each hunt group where agentID is a member – groupDn – deviceID; FQN of hunt group – groupName – string; provisioned hunt group name – makeBusy – Boolean; true (Not Ready) or, false (Ready)

Event Causes

Cause code = Normal.

OpenScape Voice Operational Notes

- Agent Not Ready event depends on proper provisioning of MHLG for Hunt Make Busy capability. Refer to Section 3.2.3, “MLHG Pilot (Group Device / ACD) and Agent BGLs” for details.
- Typical examples of when this event may be generated are:
  – The agent invokes the Hunt Make Busy ON feature on the telephone. The subscriber requires explicit access to this feature at the device level using MLHG member configuration.

  - The agent invokes Set Agent State service with requested state Not Ready. The subscriber requires explicit access to this feature at the device level using MLHG member configuration.
  - The agent becomes not ready using the Auto Busy feature. If this feature is active, the agent is made Not Ready by OpenScape Voice if the agent does not answer distributed calls.
  - The craftsperson activates the Hunt Make Busy feature using configuration.
  - The agent transitions out of Busy state and its Hunt make Busy status = ON.

- An administrable CSTA application interface option named “Enabled Extended Agent States” must be enabled extendedMakeBusyStatusList privateData presentation.
- See Part II, Appendix A, “CSTA Call Scenarios”, for more information.

4.11.2.3 Agent Ready

The Agent Ready event indicates that an agent has entered the Ready state.

Event Parameters

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
agentDevice	Y	The device ID specifies the directory number of the agent device that is Ready.
agentID	N	Indicates the agent identifier.
acdGroup	N	Indicates the ACD group. Agent is Ready for all groups where it is a member.
cause	Y	Indicates a reason for the event. Always Normal.
privateData	Y	extendedMakeBusyStatusList: • extendedMakeBusyStatus - repeated for each hunt group where agentID is a member – groupDn – deviceID; FQN of hunt group – groupName – string; provisioned hunt group name – makeBusy – Boolean; true (Not Ready) or, false (Ready)

Event Causes

Cause code = Normal.

OpenScape Voice Operational Notes

- Agent Ready event depends on proper provisioning of MHLG for Hunt Make Busy capability. Refer to Chapter 3 for details.
- Typical examples of when this event may be generated are:
  - An agent auto-work timer expires. (This is a configuration option also known as post-call timer.)
  - An agent invokes the Hunt Make Busy OFF on the telephone.
  - The agent invoked Agent Ready by using the Set Agent State service.
  - The agent transitions out of Busy state and its Hunt make Busy is OFF
  - The craftsperson deactivates the Hunt Make Busy feature using configuration.
- An administrable CSTA application interface option named “Enabled Extended Agent States” must be enabled extendedMakeBusyStatusList privateData presentation.
- See Part II, Appendix A, “CSTA Call Scenarios”, for more information.

4.11.2.4 Agent Working After Call

The Agent Working After Call event indicates that an agent has entered the Working After Call state. In this state an agent is no longer connected to a call distributed from the group device but is still occupied with work related to a previously distributed call. In this state, an agent cannot receive another call distributed from an associated group device. The agent may be performing administrative duties (for example, updating a business order form) for a previous call.

Event Parameters

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
agentDevice	Y	The device ID specifies the directory number of the agent device.
acdGroup	Y	This parameter is provided when the event is associated with a group device where the Post Call Timer expired.
pendingAgentState	Y	Indicates the agent state that the agent transitions to after the agent state is no longer WorkingAfterCall. Supported states: • Not Ready • Ready • Null
agentID	N	Indicates the agent identifier.
cause	Y	Indicates a reason for the event. Always AutoWork.
privateData	Y	extendedMakeBusyStatusList: • extendedMakeBusyStatus - repeated for each hunt group where agentID is a member – groupDn – deviceID; FQN of hunt group – groupName – string; provisioned hunt group name – makeBusy – Boolean; true (Not Ready) or, false (Ready)

Event Causes

Cause code = Normal.

OpenScape Voice Operational Notes

- Agent Working After Call event depends on proper provisioning of MHLG for Post Call Timer capability. Refer to Chapter 3, “Administration and Management Considerations for CSTA” for details.
- This event occurs when an agent completes a call distributed from an MLHG provisioned with Post Call Timer capability. Post Call Timer is not supported for MLHG in Manual (Application controlled distribution).

- An administrable CSTA application interface option named “Enabled Extended Agent States” must be enabled extendedMakeBusyStatusList privateData presentation.
- See Part II, Appendix A, “CSTA Call Scenarios”, for more information.

4.11.2.5 Call Back Event

The Call Back event indicates that a call back feature has been set or cancelled between two devices.

Event Parameters

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
originatingDevice	Y	Indicates the DeviceID of the originating device when the call back relationship was established.
targetDevice	Y	Indicates the DeviceID of the target device when the call back relationship was established. This parameter is empty if all callbacks are canceled.
callBackSetCancelled	Y	Indicates whether a call back was set or cancelled. Supported values: • FALSE - Call Back was cancelled. • TRUE - Call Back was set.

OpenScape Voice Operational Notes

- The originatingDevice and targetDevice must be provisioned for callback service.
- Multiple Call Back Events are NOT sent after Cancel Call Back - All service requests. Application must assume that the call back list has been cleared after receiving the Cancel Call Back response.
- The targetDevice must be a registered subscriber on the OpenScape Voice or addressable, for example, via a SIP-Q gateway.

4.11.2.6 Do Not Disturb

The Do Not Disturb event indicates that the do not disturb feature has been changed for a device.

The privateData named DNDResume has been introduced in the DoNotDisturbEvent. The DNDResume identifies the remaining number of minutes the DND feature will stay active.

When the DNDResume time expires the OSV DND feature is turned off for the specified device and the Do Not Disturb Event is presented.

The presentation of the DNDResume parameter is controlled by an OSV provisioned CSTA Interface Option.

Event Parameters

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
device	Y	Indicates the device where the do not disturb feature was changed.
doNotDisturbOn	Y	Specifies the current state of DND. Supported values: • TRUE = Do Not Disturb is active • FALSE = Do Not Disturb is inactive
privateData	Y	The DNDResume parameter is supported to specific time duration for the DND feature to remain active. This parameter is an integer type value. The rules for presentation are as follows: • If the doNotDisturbOn=FALSE then the DNDResume parameter is presented as null parameter. • If the doNotDisturbOn=TRUE then the DNDResume contains a value of 0-1440 minutes, which represents the remaining snooze time.

XML Example - CSTA over TCP

```
<?xml version="1.0" encoding="UTF-8"?>
<DoNotDisturbEvent xmlns="http://www.ecma-international.org/
standards/ecma-323/csta/ed4">
  <monitorCrossRefID>1150391655</monitorCrossRefID>
  <device>
    <deviceIdentifier>+15619231439</deviceIdentifier>
  </device>
  <doNotDisturbOn>true</doNotDisturbOn>
   <extensions>
    <privateData>
```

```
      <private xmlns:scx="http://www.siemens.com/schema/csta">
        <scx:DNDResume>60</scx:DNDResume>
      </private>
    </privateData>
  </extensions>
</DoNotDisturbEvent>
```

XML Example - CSTA over SIP

```
<?xml version="1.0" encoding="UTF-8"?>
<DNDE xmlns="http://www.ecma-international.org/standards/ecma-
323/csta/ed4">
  <mCRI>1150391655</mCRI>
  <dvc>
    <dIr>+15619231439</dIr>
  </device>
  <dNDO>true</dNDO>
   <extn>
    <pDa>
      <prvt xmlns:scx="http://www.siemens.com/schema/csta">
        <scx:zDRm>60</scx:zDRm>
      </prvt>
    </pDa>
  </extn>
</DNDE>
```

OpenScape Voice Operational Notes

- If the device is an internal MLHG (Hunt Group) the event provide status of the Night Service overflow for the group.
- To synchronize OpenScape Voice centralized Call Forwarding with an OpenStage the CSTA device category must be set to “CSTA over SIP”. If multiple device contacts are registered with uaCSTA, then the DND event is sent to all contacts.

4.11.2.7 Forwarding

The Forwarding event indicates that the forwarding feature has been changed for a device.

Event Parameters

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
device	Y	Indicates the device where the forwarding feature was changed.
forwardingType	Y	Indicates the type of forwarding. Supported values: • forwarding Immediate • fowardBusy • fowardNoAns • forwardBusyInt • forwardBusyExt • forwardNoAnsInt • forwardNoAnsExt • forwardImmInt • forwardImmExt The “internal” and “external” types are lower precedence types that refer to the type of call origination (for example an external call) that will be forwarded if it matches a forwarding type (for example forwardImmExt) enabled at the device provided that the higher precedence type (e.g.forwardImmediate) had not already forwarded the call.
forwardStatus	Y	Indicates the status of the forwarding type. Supported Values: • FALSE - the forwarding type is deactivated • TRUE - the forwarding type is active
forwardTo	Y	Specifies the destination to which calls are forwarded. It is only provided when call forwarding is being activated.
ringCount	N	Specifies the number of times a device should ring prior to forwarding no answer.
ringDuration	N	Specifies the time (in seconds) that the device should ring prior to forwarding no answer. ringDuration should not be provided if ringCount is provided.
privateData	Y	The following privateData elements are supported: • forwardingCapsChange - Boolean; true=changed, false=no change

OpenScape Voice Operational Notes

- CSTA exposes OpenScape Voice Call Forwarding status for Immediate, No Answer and Busy variants only.

- An administrable CSTA application interface option named “Enable CF Privilege Change Notification” must be enabled to forwardingCapsChange privateData presentation.
- To synchronize OpenScape Voice centralized Call Forwarding with an OpenStage the CSTA device category must be set to “CSTA over SIP”. If multiple device contacts are registered with uaCSTA, then the forwarding event is sent to all contacts.

4.12 Device Maintenance

4.12.1 Events

OpenScape Voice does not support the Partially in Service event.

The following table illustrates actions that may cause the transmission of a CSTA Back in Service or Out of Service event:

Action	Device State	Event transmitted
An application starts monitor on device (DN)	Registered	None *
	Deregistered	None *
Monitored device (DN) is blocked	Blocked	Out of Service ***
Monitored device (DN) is deleted using administration.	Deleted	None (no existing monitor)
Monitored device (DN) is unblocked	Unblocked	None (no existing monitor)
Monitored device (DN) registers with OpenScape Voice or, branch office transitions out of survivable mode.	Registered	Back in Service **
Monitored device (DN) deregisters with OpenScape Voice	Deregistered	Out of Service **
Monitored device (DN) is suspended	Registered	Out of Service **
An application or OpenScape Voice stops monitor on device (DN)	Last known state	None
Device is placed in a suspended state. For example, if branch office enters survivable mode.	Suspended	Out-of-service
* The application may send a Snapshot device to determine device maintenance state. OpenScape Voice returns an error code in the Snapshot device response if the device is provisioned and not in service, unless device is provisioned only (for example, virtual device). ** OpenScape Voice uses device state transition hysteresis to avoid performance problems by generating too many unnecessary events. *** Subscriber is no longer valid for CSTA Services.		

Special considerations for ONS subscribers:

The following describes the handling requirements for the OutOfService, BackInService events and the Snapshot Device response as it relates to ONS subscribers:

The privateData element; **outOfService** status is presented when no SIP devices are register for a subscriber's ONS.

OSV currently sends the CSTA OutOfService device maintenance event when there are no longer any SIP devices registered for the ONS subscriber DN. Once the first SIP device registers for the ONS DN CSTA sends a BackInService device

maintenance event. If additional SIP devices register for this same in-service device CSTA does not repeat BackInService events for the subsequent device registrations.

Applications should already be familiar Snapshot Device Response enhancement for ONS subscriber DN.

The following table illustrates how the application should handle the Snapshot Device and Device Maintenance Events:

ONS Device Registration Transition	Active ONS Call(s)	Snapshot Device Response	Device Maintenance Event
Registered -> Not Registered	No active calls	Standard CSTA Error = outOfService (NAK)	OutOfService Event **
	Call(s) with ONS != OND	Standard response with call list and privateData; outOfService = true	
Not Registered -> Registered	No active Calls	Standard response with call list and privateData; outOfService = true	BackInService Event
	Call(s) with ONS=OND and/or ONS != OND	Standard response (ACK) Standard response; with call list; privateData; outOfService = false	

- Not registered means that there are no SIP contacts currently registered for the ONS subscriber DN. If more than one SIP contact is registered for the ONS subscriber DN then the ONS is considered registered and in-service.

****This OOS event means that only the physical ONS registering device is Out-Of-Service. The ONS subscriber DN may still be used to provide binding with other internal or external ONDs, such as OSMO.**

4.12.1.1 Back in Service

The Back in Service event indicates that the device has been returned to service and is operating normally.

Event Parameters

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
Device	Y	Indicates the device that is back in service.
Cause	Y	Specifies a reason for the event. Always Normal.

Event Causes

Cause code = Normal.

OpenScape Voice Operational Notes

- This event is sent when the SIP device registers or transitions from suspended mode. In the case of multiple devices registrations for the same DN, this event is sent when the first contact for the DN registers.
- Not provided for Group Device (MLHG Pilot).
- The Back in Service event does not imply that the capabilities of the device out of service have changed.

4.12.1.2 Device Capabilities Changed

The Device Capabilities Changed event indicates that device level information that can be obtained with the capability exchange services (for example, Get Logical Device Information services) has changed.

The current device level capability information can be obtained by issuing a Get Physical Device Information and/or a Get Logical Device Information service.

Event Parameters

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
device	Y	Indicates the device whose information has changed. See usage notes below.
cause	Y	Specifies a reason for the event

Event Causes

Cause code = Normal.

OpenScape Voice Operational Notes

- The device capabilities event is sent by OpenScape Voice in the following case(s):
  - When a member is added to a Group Device (MLHG) Pilot DN using administration.
  - When a member is removed from a Group Device (MLHG) Pilot DN using administration.
- The subjectDeviceID is the Group Device Pilot DN.
- The Device Capabilities Changed event shall be generated for the above condition whether or not the device level capability information has been previously obtained using the capability exchange services.

4.12.1.3 Out of Service

The Out Of Service event indicates that the device has entered a maintenance state (that is, has been taken out of service) and can no longer accept calls and some categories of CSTA service requests (Call Control services, for example).

Event Parameters

Parameter Name	Supported	Comments
monitorCrossRefID	Y	Associates the event to an established monitor.
Device	Y	Indicates the device that has been taken out of service.
Cause	Y	Specifies a reason for the event, always event cause'Normal'

Event Causes

Cause code = Normal.

OpenScape Voice Operational Notes

- Event is reported when SIP device de-registers or enters suspended mode. In the case of multiple devices, registrations for the same DN event are sent only when the last contact for the DN un-registers.
- Event reported when a SIP device behind a proxy enters the "suspended" state. The Suspended state is detected when the associated proxy is not reachable (e.g. WAN connection is down, Proxy failure, Call Admission Control restriction, etc.).
- When a device goes out of service, existing monitors are not removed, existing MonitorCrossRefIds remains valid.
- Snapshot services results in a negative acknowledgement (error code: Device Out of Service) if attempted on a device out of service.

4.13 I/O Services

Service Not Supported

I/O-Services support the exchange of data between a computer application (a computing function component) and a telephony device (to send Data from the computer application to the display of a telephony device, or to send Data from the keypad of a telephony device to the computer application, etc.).

4.14 Vendor Specific Extensions Service

Service Not Supported

CSTA allows the provision of value added services and events that are beyond what is defined in this Standard. It is possible both to extend the existing services and events defined in this Standard as well as to create completely new services and events. A vendor may choose to support a vendor specific extension with the understanding that it may not interoperate with other CSTA (Phase III)-compliant products.

4.15 Data Collection Services

Service Not Supported

The Data Collection services are used to collect information such as DTMF/rotary pulse digits and Telephony Tones that is received by a device over a connection.

4.16 Voice Services and Events

Service Not Supported

A Voice Unit device allows messages consisting of voice stream data to be created, manipulated, played to a Connection, or recorded from a Connection. A Voice Unit device can be observed and controlled using the CSTA Voice services.

5 Internal Service Framework

OpenScape Voice provides a framework that enables the implementation of internal CSTA-base applications developed using C++.

The internal service framework is designed to run on the OpenScape Voice server cluster in a geo-separated environment to enable the internal applications to support failover and recovery of persistent data, such as Call Log entries and to allow for synchronization of the data between nodes.

5.1 Centralized Call Log (CCL) Service

The first service to be developed under the new framework is the Centralized Call Log service.

The Centralized Call Log service is designed to enable WebRTC clients, such as Ansible Connector Clients, to discover their subscriber's call log data, track call log events, add (local data) and delete call log entries from a common database in the switching function.

After startup clients send the Call Log Snapshot request which initiates the sending of subsequent call log events from the switching function. Clients may also send requests to the switching function to delete all or specific call log entries from the database.

5.1.1 Private Namespace (non-ECMA namespace)

In order for OSV to support pass through of non-ECMA 323 messages a new xml schema has been introduced to identify the new services, events and errors created for an OSV internal service.

The proposed namespace will have the basic format xmlns = http://www.unify.com/internal-services

- For Call Log namespace: xmlns = http://www.unify.com/internal-services/call-log/ed1/
- Internal Service Error namespace: xmlns = http://www.unify.com/internal-services/internal-service-errors/ed1
- The most current schema versions are stored in g-dms at https://rcus.g-dms.com/livelink/livelink.exe?func=ll&objId=59844129&objAction=browse&sort=name&viewType=1
- In the future other internal-services schemas may be defined.

When OSV observes a service request with a name space that is not part the ECMA 323 (ECMA 269) or ECMA 354 (Application Session Services) namespace it knows to pass these services to the internal service message distributor.

5.1.2 XML Short Tags Definition

The Call Log Service and Event parameters are described in the following sections using their long name format. However, only short tags are presented via on the CSTA over SIP and CSTA over TCP interfaces.

The Call Log services and events short xml tags are shown in the table below. The CF must be able to use and identify the Call Log short tag formats listed below:

Long Parameter Names	Proposed short XML format
CallLogSnapshot	**CLS**
• subscriber	• sub
CallLogSnapshotResponse	**CLSR**
CallLogDeleteEntry	**CLDE**
• callLogDeleteList	• cLDL
• recordID	• rID
CallLogDeleteResponse	**CLDR**
CallLogEvent	**CLE**
• subscriber	• sub
• callLogList	• cLL
• callLogEntry	• cLE
– recordID	– rID
– callID	– cID
– recordType	– rTp
– createTime	– cTm
– startTime	– sTm
– callDuration	– cDr
– callingDevice	– clDv
– calledDevice	– cldDv
– answeringDevice	– aDv
– triggeringDevice	– tDv
– redirectedToDevice	– rTDv
– result	– res
– characteristics	– char
– confList	– cfL
• confMemberDetails	• cfMd
confMemberNumber origStartTime ansDuration confStartTime confDuration	cfMn oSTm aDur CfSTm cfDr
• callLogDeleteList	• cLDL

Long Parameter Names	Proposed short XML format
– recordID	– rID
InternalServiceErrorCode	**ISEC**
• code	• code

5.1.3 Centralized Call Log Services and Events

Call Log Snapshot, Call Log Delete service and Call Log Event are new internal CSTA services introduced by OpenScape Voice. These internal services and events are transported on a standard OSV CSTA connection and are identified by a private namespace. These new services and events are transported primarily via CSTA over SIP to support CSTA-enabled client, such as; Ansible Connector Clients. But these services are also supported via CSTA over TCP interface as well. There is currently no private WSDL support for these services and events via CSTA over HTTP/SOAP.

To ensure backward compatibility OSV management has introduced a CSTA over TCP interface option to enable / disable Call Log services and events.

5.1.3.1 CSTA Call Control Event Tracking

The following describes basic rules for CSTA event tracking used by OpenScape Voice to create, update and finalize call log entries:

- Delivered - Creates entry with basic information, e.g., record type, calling and calling device, create time.
- Failed - Creates or updates entry to indicate unsuccessful result, e.g., busy.
- ConnectionCleared - Creates or finalizes entry with previously determined result, answered, notAnswered, busy, etc.
- Diverted - Finalizes entry for called device because of a call diversion, e.g., Call Forwarding.
- Established - Creates or updates final information for an entry.
- Held and Retrieved - Creates and updates entry for a silent monitoring device.
- CallInformation - Update entry information, e.g., video characteristics and in a FUTURE release provides answer notification of a previously diverted call.
- Conferenced - Identifies new callID to enable internal service to create and update the entry's conference list.
- Transferred - Identifies new callID to enable internal service to correctly process subsequent events and identify transfer result in affected entries.

5.1.3.2 Call Log Services Description - Call Log Snapshot (CF ->SF)

The Call Log Snapshot service is a proprietary service that is used to initiate call logging events for the specified device. After the request is acknowledged the internal Call Log application shall send a Call Log Event with the current list of call log entries.

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model. A Call Log Snapshot Response will be sent to the CF with no parameters. See Call Log Error Codes below.

Service Request Parameters

Parameter Name	Type	Comments
subscriber	deviceID	Specifies the serviceID of OSV subscriber for whom Call Logging event flow should be enabled. Mandatory for CSTA over TCP.

Service Response Parameters

Call Log Snapshot Response has no parameters.

Internal Service Error Codes

Call Log Error Code(s) shall be sent under the following conditions:

Internal Service Error Code	Condition
subscriberNotFound	The OSV subscriber ID was not located. No call log event will be returned for this subscriber.
dispatcherNotActive	The CSTA dispatcher service in not active. Try again.
internalResourceBusy	Service request could be handled at this time, e.g., Time out. Try again
invokeIdAlreadyInUse	The specified invokeID is in use. Try again.

NOTES:

The Web-RTC client shall provide a retry mechanism if no response is received within reasonable amount of time.

XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<CLS xmlns="http://www.unify.com/internal-service/call-log/ed1">
 <sub>+15610231439</subscriber>
</CLS>

<?xml version="1.0" encoding="UTF-8"?>
<CLSR xmlns="http://www.unify.com/internal-service/call-log/ed1" />

ERROR Example:
<?xml version="1.0" encoding="UTF-8"?>
<ISEC xmlns="http://www.unify.com/internal-service/ed1">
  <code>subscriberNotFound</code>
</ISEC>
```

5.1.3.3 Call Log Services Description - Call Log Delete Entry (CF -> SF)

The Call Log Delete entry service is a proprietary service that is used to delete a call log entry from the centralized database. After the service request is acknowledged a Call Log Event is sent to confirm the deletion of the specified records.

Acknowledgement Model

OpenScape Voice supports the atomic positive acknowledgement model. A Call Log Delete Response will be sent to the CF with no parameters.

<table>
<tr><th>Parameter Name</th><th>Type</th><th>Comments</th></tr>
<tr><td>subscriber</td><td>deviceID</td><td>Specifies the serviceID of the OSV subscriber where call log event shall be enabled. Mandatory for CSTA over TCP.</td></tr>
<tr><td>callLogDeleteList</td><td>list</td><td>Specifies the list of unique call log recordID to be deleted.
One or more recordIDs may be provided. NOTE: If no recordID parameter is provided then all records shall be deleted for the specified subscriber.
<table>
<tr><th>Parameter Name</th><th>Type</th><th>Comments</th></tr>
<tr><td>recordID</td><td>string</td><td>specifies the unique call log record ID to be deleted from SF database</td></tr>
</table></td></tr>
</table>

Service Response Parameters

Call Log Delete Response has no parameters.

Internal Service Error Codes

A Call Log Error Code shall be sent under the following conditions:

Internal Service Error Codes	Condition
subscriberNotFound	The OSV subscriber ID was not located. No call log event will be returned for this subscriber.
invalidRecordID	The recordID format was invalid, for example, null recordID.
dispatcherNotActive	The CSTA dispatcher service in not active. Try again.
internalResourceBusy	Service request could be handled at this time, e.g., Time out. Try again
invokeIdAlreadyInUse	The specified invokeID is in use. Try again.

Note: The CF application must provide a retry mechanism if no response is received within reasonable amount of time.

xml short tag format

```
<?xml version="1.0" encoding="UTF-8"?>
<CLD xmlns="http://www.unify.com/internal-service/call-log/ed1">
 <sub>+15610231439</subscriber>
 <cLDL>
  <rID>R12</delRecordID>
  <rID>R13</delRecordID>
 </cLDL>
</CLD>

<?xml version="1.0" encoding="UTF-8"?>
<CLDR xmlns="http://www.unify.com/internal-service/call-log/ed1" />
```

5.1.3.4 Call Log Event Description - Call Log Event (SF -> CF)

The Call Log Event is generated by OSV internal Call Log service (CCL). This event that is used to update a client's local call log data.

The CF must send Call Log Snapshot service request in order to initiate the flow of Call Log events.

The Call Log Event is presented in following cases:

- After a client invokes a Call Log Snapshot service request and the Call Log application acknowledges the request. The event will contain a list of all available call log entries for the specified subscriber.

  Note: Depending on the final size of a callLogEntry it may be necessary for CCL send several Call Log Events if the size for the complete call log list is too large for a single record. To build their local call log data table, clients must be designed to support multiple Call Log Events with multiple entries.

- After a client invokes a Call Log Delete service request and the Call Log application acknowledges the request. The event will contain a list of all call log entries that were deleted from the database for the subscriber.

- When a new call log entry has been added to the database for the subscriber. The callLogList will contain the new entry and if included the callLogDeletedEntryList may contain the recordID of a call log entry that was overwritten do the maximum number of entries.

Event Parameters

Parameter Name	Type	Comments
subscriber	deviceID	Specifies the serviceID of OSV subscriber where call log event shall be enabled. Mandatory for CSTA over TCP.
callLogList	List of structures	One or more callLogEntry structures. ***NOTE: The callLogList, confList and callLogDeleteList include the "entries" attribute. This attribute specifies the number of entries in each list.***

Parameter Name	Type	Comments
callLogEntry	**structure**	One complete call log entry is repeated for each available for specified subscriber.
recordID	string	Unique call log record ID.
callID	csta:callID	OSV CSTA callID associated with the call log entry.
recordType	string	Call direction relative to the specified subscriber • inbound • outbound • joined
createTime	integer	Call log creation time **
startTime	integer	Call log start time (answer) **
callDuration	integer	Call duration in seconds – based on create time if not answered or start time if answered.
callingDevice	csta:callID	Calling deviceID *
calledDevice	csta:callID	Called deviceID *
answeringDevice	csta:callID	Answering deviceID *

Parameter Name	Type	Comments		
		triggeringDevice	csta:callID	Triggering deviceID * • In redirection (DivertedEvent) this is the redirection device • In the case of transfer this is the transferring device • In the case of conference this conferencing device
		redirectedToDevice	csta:callID	Redirected-To deviceID *
		result	string	String value that indicates the disposition of call. Possible values are: answered – call was answered by the originally called device or automatically answered by a device that is silently monitored. answeredElsewhere – call was answered by another device, for example after call forwarding. notAnswered – the calling device released before the called device answered. busy – call failed because called device reported busy. notKnown – call failed because called device reported an error other than busy. answeredByVoiceMail – call was redirected and answered by the voice mail on behalf of the originally called device. redirected – call was redirected, for example forwarding, deflect call, pickup, etc. transferred – indicates that the reported call was transferred. A new call log entry will be created for the calling and called devices in a new connection. conferenced – indicates that the calling and called device were joined in a station controlled conference. This also applies to Keyset Bridging and Silent/Active monitoring. The confList parameter provides the information about the other conference members.
		characteristics	bitmap	One of the following boolean values that identifies the following characteristics of the call: • voiceOnly • video
		confList	list of structures	
		confMemberDetails	**structure**	
		confMemberNumber origStartTime ansDuration confStartTime confDuration	csta:callID integer integer integer integer	Conference member's deviceID* Original start time ** Answer duration - seconds Conference start time Conference duration
		* Reduced SFR format. Includes NM field tags are presented, such as, Name, ond tag and keyID tag are presented. No other		

<table>
<tr><th>Parameter Name</th><th>Type</th><th>Comments</th></tr>
<tr><td></td><td></td><td>tags such as, uid, epid, etc. are not presented. For example: N<FQDN in GNF format>name;ond=GNF;keyID=GNF

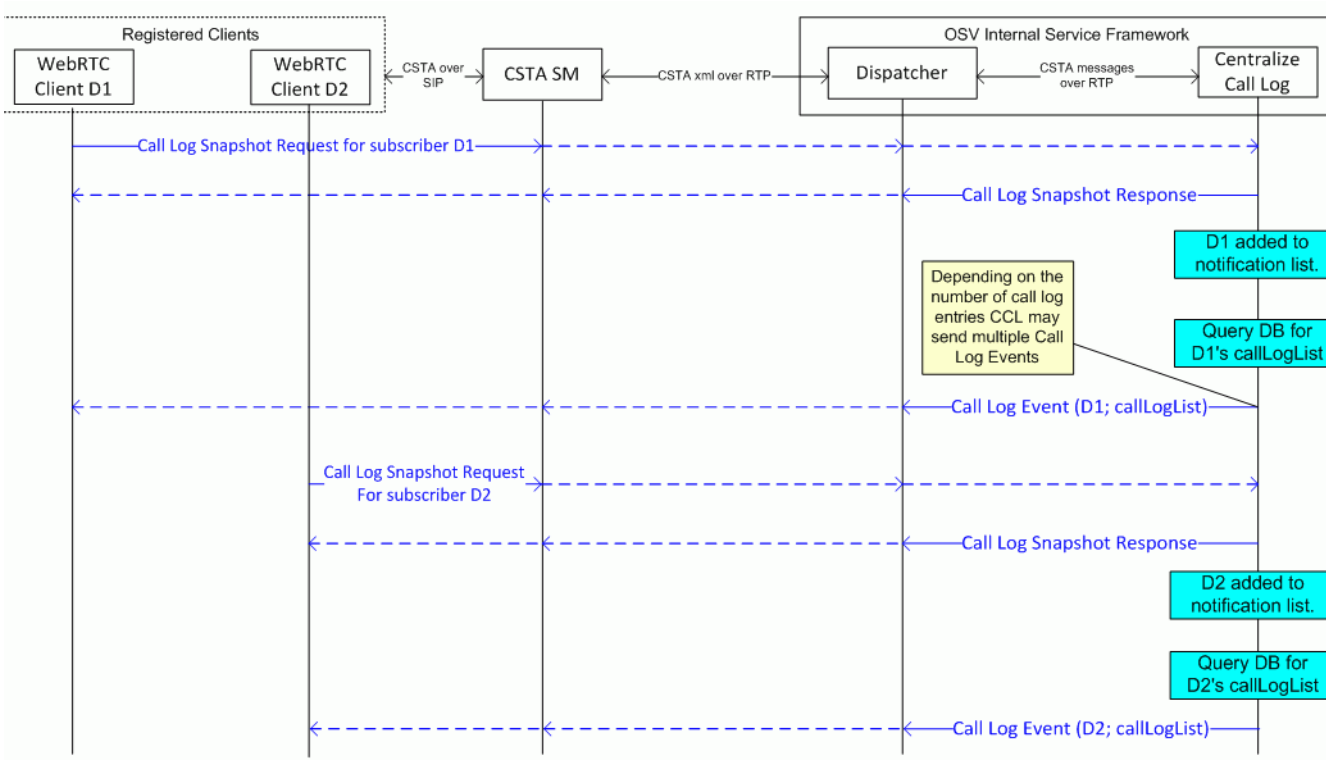
** Timestamp (based on system time) that call log record was created. Presented as seconds since January 1st 1970 (Time_T).</td></tr>
<tr><td>callLogDeleteList</td><td>List</td><td>A list of one or more recordID(s) that have been deleted from the SF database:
<table>
<tr><th>Parameter Name</th><th>Type</th><th>Comments</th></tr>
<tr><td>recordID</td><td>string</td><td>specifies the unique call log record IDs deleted from SF database</td></tr>
</table></td></tr>
</table>

xml short tag example

```
<?xml version="1.0" encoding="utf-16"?>
<CLE xmlns="http://www.unify.com/internal-service/call-log/ed1">
  <sub>+15619231758</subscriber>
  <cLL e=2>
   <cLE>
    <rID>R21-1</rID>
    <rTp>inbound</rTp>
    <cTm>12345672</cTm>
    <sTm>12345678</sTm>
    <cDr>640</cDr>
    <clDv>N&gt+15619231439&lt;Gary Mora;ond=9931439</clDv>
    <cldDv>N&gt+15619231575&lt;Jeff Cripe</cldDv>
    <aDv>N&gt+15619231575&lt;Jeff Cripe</aDv>
    <res>answered</res>
    <char>video</char>
   </cLE>
   <cLE>
    <rID>R21-1</rID>
    <rTp>inbound</rTp>
    <cTm>12345678</cTm>
    <sTm>12345678</sTm>
    <cDr>640</cDr>
    <clDv>N&gt+15619231439&lt;Gary Mora;ond=9931439</clDv>
    <cldDv>N&gt+15619231575&lt;Jeff Cripe</cldDv>
    <tDv> N&gt+15619231575&lt;Jeff Cripe <tDv>
    <rTDv>+15619231999</rTDv>
    <res>redirected</res>
    <char>voiceOnly</char>
   </cLE>
</cLL>
```

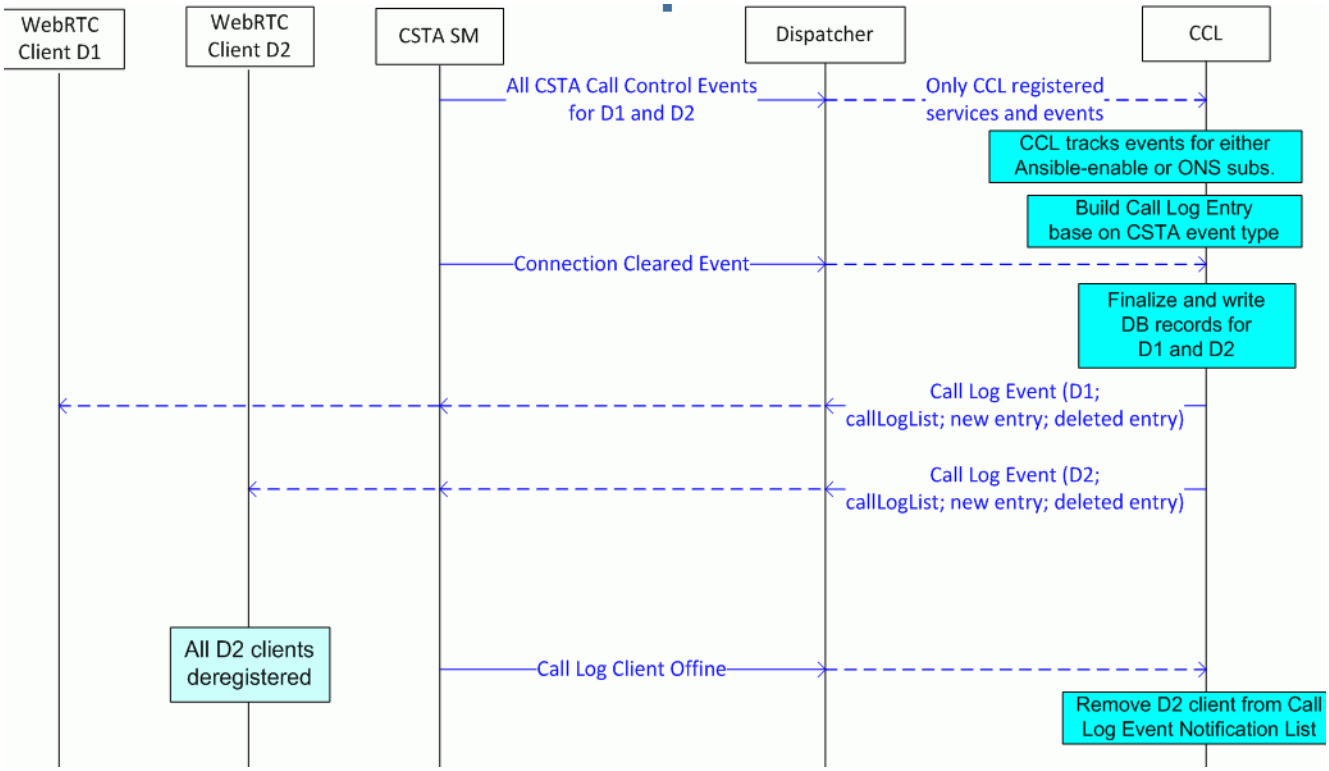
5.1.3.5 Event Flow Examples

Call Log Discovery Flow

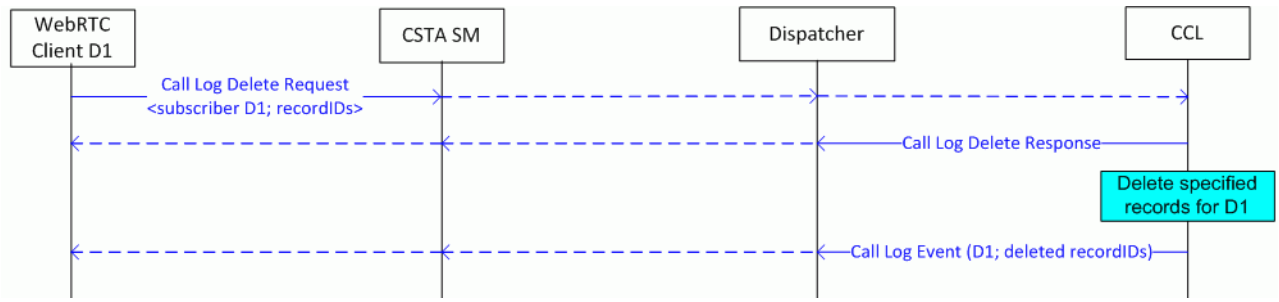


Note: If all contacts of a client de-register then the deviceID is removed from the notification and Call Log Events are suspended until the client reregisters and snapshots call log data.

Call Log Real Time Update Flow



Call Log Delete Flow



A CSTA Call Scenarios

PART II - CSTA Event Flows: are mostly written in the format of ECMA TR/82 (ECMA TR/82 Scenarios for Computer Supported Telecommucation Applications (CSTA) Phase III), and describe specific call flows between a CSTA-enabled application and OpenScape Voice.

The Call scenarios are Excel files that make up this part of the document. They are Part II of the CSTA Interface Specification and are located in a zip file under *OpenScape Voice V9 V9 Interface Manual: Volume 4, CSTA Interface*.

B OpenScape Voice privateData Schema

OpenScape Voice V9 uses ECMA-323 edition 4.

PART III - CSTA Private Data Schema: The published OpenScape Voice V9 privateData schema are files that make up this part of the document. They are Part III of the CSTA Interface Specification and are located in a zip file under *OpenScape Voice V7 Interface Manual: Volume 4, CSTA Interface.*

B.1 OpenScape Voice Private Data Schema - ECMA 323 and ECMA 354

OpenScape Voice V3.1 and higher - exposes ECMA 323 edition 4 and ECMA 354 edition 1.

The published privateData schema is available in E-Doku by downloading the most current OpenScape Voice, Interface Manual: Volume 4 .zip file.

B.2 OpenScape Voice Internal Services Schema

B.2.1 Centralized Call Log

The published privateData schema is available in E-Doku by downloading the most current OpenScape Voice, Interface Manual: Volume 4 .zip file.

C Supported CSTA Services and Events by Device Type

The following table identifies supported CSTA services and events for CSTA over TCP and CSTA over SIP interfaces.

Station, Group and Network Device Types (CSTA over TCP interface):

Normal = Basic CSTA type for any SIP phone that does not support auto-answer

Type1 = CSTA type for OpenStage and OptiPoint phones that support auto-answer and keyset operation

Hunt Group = internal OSV hunt group provisioned for CSTA services. CSTA services are only allowed when hunt group is provisioned for Application Controlled hunting method.

Network Device (NID) = A SIP gateway where the endpoint profile is provisioned with CSTA services

Client Device Types (CSTA over SIP interface):

OSMO = OpenScape Mobile client supported services and events

OpenStage uaCSTA = Basic uaCSTA supported services and events

Applicable to CSTA over TCP and CSTA over SIP interfaces:

ONS = One Number Service

CSTA Services and Events	CSTA over TCP	Station, Group and Network Device Types				CSTA Over SIP	Client Device Types		ONS Support
		Normal	Type1	Hunt Group	Network Devices		OSMO	OpenStage (uaCSTA)	
CAPABILITIES EXCHANGE SERVICES									
Services									
get CSTA features (C → S)	Yes	✓	✓	✓	✓	Yes			
get logical device Information (C → S)	Yes	✓	✓	✓	✓	Yes			
get physical device information (C → S)	Yes	✓	✓			Yes			
get switching function capabilities (C → S)	Yes	✓	✓	✓		Yes			
get switching function devices (C → S)	Yes			✓	✓	No			
switching function devices (S → C)	Yes			✓	✓	No			
get configuration data (CSTA over SIP only)	No					Yes	✓ [1]		

Table 13 Supported CSTA Services and Events (Seite 1 von 11)

CSTA Services and Events	CSTA over TCP	Station, Group and Network Device Types				CSTA Over SIP	Client Device Types		ONS Support
		Normal	Type1	Hunt Group	Network Devices		OSMO	OpenStage (uaCSTA)	
SYSTEM SERVICES									
Registration Services									
change system status filter (C → S)	No					No			
system register (C → S)	Yes					Yes			
system status register abort (S → C)	No					No			
system status register cancel (C → S)	No					No			
Services									
request system status (C ←→S)	Yes					Yes			
system status (C ←→S)	Yes					Yes			
switching function capabilities changed (S → C)	No					No			
switching function devices changed (S → C)	No					No			
MONITORING SERVICES									
change monitor filter (C → S)	Yes	✓	✓	✓	✓	No	✓ [2]		
monitor start (C → S) (device monitor only)	Yes	✓	✓	✓	✓	No	✓ [3]		
monitor stop (C ←→ S) (device monitor only)	Yes	✓	✓	✓	✓	No	✓ [4]		
SNAPSHOT SERVICES									
Services									
snapshot call (C → S)	Yes	✓	✓	✓		Yes	✓		✓
snapshot device (C → S)	Yes	✓	✓	✓		Yes	✓ [5]		✓
snapshot calldata (S → C)	No					No			
snapshot device data (S → C)	Yes	✓	✓	✓		Yes	✓		✓
APPLICATION SESSION SERVICES									
Services									
start application session (C → S)	Yes					No			
stop application session (C → S)	Yes					No			
reset application session timer (C → S)	Yes					No			
application session terminated (S → C)	Yes					No			

Table 13 Supported CSTA Services and Events (Seite 2 von 11)

CSTA Services and Events	CSTA over TCP	Station, Group and Network Device Types				CSTA Over SIP	Client Device Types		ONS Support
		Normal	Type1	Hunt Group	Network Devices		OSMO	OpenStage (uaCSTA)	
CALL CONTROL									
Services (C → S)									
accept call	Yes	✓	✓			Yes			✓
alternate call	Yes		✓			Yes			✓
answer call	Yes		✓			Yes		✓	✓
call back call-related	Yes					Yes			✓
call back message call-related	No					No			
camp on call	No					No			
clear call	No					No			
clear connection	Yes	✓	✓	✓		Yes	✓ [6]	✓	✓
conference call	Yes	✓	✓			Yes			✓
consultation call	Yes	✓	✓			Yes		✓	✓
deflect call	Yes	✓	✓	✓		Yes	✓		✓
dial digits	No					No			
directed pickup call	No					No			
group pickup call	No					No			
hold call	Yes	✓	✓			Yes		✓	✓
intrude call	No					No			
join call	Yes	✓	✓			Yes			✓
make call (supports prompt and do not prompt)	Yes		✓			Yes	✓ [7]	✓	✓
make predictive call	No					No			
park call	No					No			
reconnect call	Yes		✓			Yes			✓
retrieve call	Yes	✓	✓			Yes	✓ [8]	✓	✓
send message	No					No			
single step conference call	No					No			
single step transfer	Yes	✓	✓			Yes	✓ [9]		✓
transfer call	Yes	✓	✓			Yes			✓
Events									
bridged	No					No			

Table 13 Supported CSTA Services and Events (Seite 3 von 11)

CSTA Services and Events	CSTA over TCP	Station, Group and Network Device Types				CSTA Over SIP	Client Device Types		ONS Support
		Normal	Type1	Hunt Group	Network Devices		OSMO	OpenStage (uaCSTA)	
call cleared	No					No			
conferenced	Yes	✓	✓			Yes			✓
connection cleared	Yes	✓	✓	✓		Yes			✓
delivered	Yes	✓	✓	✓		Yes			✓
digits dialed	No					No			
diverted	Yes	✓	✓	✓		Yes			✓
established	Yes	✓	✓			Yes			✓
failed	Yes	✓	✓	✓		Yes			✓
held	Yes	✓	✓			Yes			✓
network capabilities changed	No					No			
network reached	Yes	✓	✓			Yes			✓
offered	Yes	✓	✓		✓	Yes			✓
originated	Yes	✓	✓		✓	Yes	✓		✓
queued	Yes			✓		Yes	✓		✓
retrieved	Yes	✓	✓			Yes	✓		✓
service initiated	Yes	✓	✓			Yes	✓		✓
transferred	Yes	✓	✓			Yes	✓		✓
CALL ASSOCIATED FEATURES									
Services (C → S)									
associate data	Yes				✓	No			
cancel telephony tones	No					No			
change connection information	Yes	✓	✓			Yes			✓
generate digits	Yes	✓	✓			Yes		✓	✓
generate telephony tones	No					No			
send user information	No					No			
Events									
call information	Yes	✓	✓	✓	✓	Yes	✓		✓
charging	No					No			
digits generated	Yes	✓	✓			Yes		✓	✓
telephony tones generated	No					No			

Table 13 Supported CSTA Services and Events (Seite 4 von 11)

CSTA Services and Events	CSTA over TCP	Station, Group and Network Device Types				CSTA Over SIP	Client Device Types		ONS Support
		Normal	Type1	Hunt Group	Network Devices		OSMO	OpenStage (uaCSTA)	
service completion failure	Yes	✓	✓			Yes	✓		✓
MEDIA ATTACHEMENT SERVICES & EVENTS	Not Supported								
Services (C → S)									
attach media service	No					No			
detach media service	No					No			
Events									
media attached	No					No			
media detached	No					No			
ROUTING SERVICES									
Registration Services									
route register (C → S)	Yes					Yes			
route register abort (S → C)	Yes					Yes			
route register cancel (C → S)	Yes					Yes			
Services									
reroute (S → C)	Yes					Yes			
route end (C → S)	Yes					Yes			
route reject (C → S)	Yes					Yes			
route request (S → C)	Yes					Yes			
route select (C → S)	Yes					Yes			
route used (S → C)	No					No			
PHYSICAL DEVICE FEATURES									
Services (C → S)									
button press	No					No			
get auditory apparatus information	No					No			
Get button information	No					No			
Get display	No					No			
Get hook switch status	No					No			
Get lamp information	No					No			
Get lamp mode	No					No			
Get message waiting indicator	Yes	✓	✓			Yes			

Table 13 Supported CSTA Services and Events (Seite 5 von 11)

CSTA Services and Events	CSTA over TCP	Station, Group and Network Device Types				CSTA Over SIP	Client Device Types		ONS Support
		Normal	Type1	Hunt Group	Network Devices		OSMO	OpenStage (uaCSTA)	
Get microphone gain	No					No			
Get microphone mute	Yes					Yes		✓	
Get ring status	No					No			
Get speaker mute	No					No			
Get speaker volume	Yes					Yes		✓	
Set button information	No					No			
Set display	No					No			
Set hook switch status	No					No			
Set lamp mode	No					No			
Set message waiting indicator	Yes	✓	✓			Yes			
Set microphone gain	No					No			
Set microphone mute	Yes					Yes		✓	
Set ringer status	No					No			
Set speaker mute	No					No			
Set speaker volume	Yes					Yes		✓	
Events									
button information	No					No			
button pressed	No					No			
display updated	No					No			
hook switch	No					No			
lamp mode	No					No			
message waiting	Yes	✓	✓			Yes	✓		
microphone gain	No					No			
microphone mute	No					No			
ringer status	No					No			
speaker mute	No					No			
speaker volume	No					No			
LOGICAL DEVICE FEATURES									
Services (C → S)									
call back non-call-related	Yes	✓	✓			Yes			

Table 13 Supported CSTA Services and Events (Seite 6 von 11)

CSTA Services and Events	CSTA over TCP	Station, Group and Network Device Types				CSTA Over SIP	Client Device Types		ONS Support
		Normal	Type1	Hunt Group	Network Devices		OSMO	OpenStage (uaCSTA)	
call back message non-call-related	No					No			
cancel call back	Yes	✓	✓			Yes			
cancel call back message	No					No			
get agent state	Yes	✓	✓			Yes			
get auto work mode	No					No			
get caller ID status	No					No			
get do not disturb (* for Night Service Status)	Yes	✓	✓	✓ *		Yes		✓	
get forwarding	Yes	✓	✓			Yes	✓	✓	
Get last number dialed	No					No			
Get routing mode	No					No			
Set agent state	Yes	✓	✓			Yes			
Set auto answer mode	No					No			
Set auto work mode	No					No			
Set caller ID status	No					No			
Set do not disturb (* for Night Service Status)	Yes	✓	✓	✓ *		Yes		✓	
Set forwarding	Yes	✓	✓			Yes	✓	✓	
Set routing mode	No					No			
Events									
agent busy	Yes	✓	✓			Yes			
agent logged off	No					No			
agent logged on	No					No			
agent not ready	Yes	✓	✓			Yes			
agent ready	Yes	✓	✓			Yes			
agent working after call	Yes	✓	✓			Yes			
auto answer	No					No			
auto work mode	No					No			
callback event	Yes	✓	✓			Yes			
callback message	No					No			
caller ID status	No					No			
do not disturb	Yes	✓	✓	✓		Yes		✓	

Table 13 Supported CSTA Services and Events (Seite 7 von 11)

CSTA Services and Events	CSTA over TCP	Station, Group and Network Device Types				CSTA Over SIP	Client Device Types		ONS Support
		Normal	Type1	Hunt Group	Network Devices		OSMO	OpenStage (uaCSTA)	
forwarding	Yes	✓	✓			Yes	✓	✓	
routing mode	No					No			
DEVICE MAINTENANCE EVENTS									
Events									
back in service	Yes	✓	✓	✓		Yes	✓		✓
out of service	Yes	✓	✓	✓		Yes	✓		✓
device capability changed	Yes			✓		Yes			
device partially in service	No					No			
I/O SERVICES	Not Supported								
Registration Services									
I/O register (C → S)	No					No			
I/O register abort (S → C)	No					No			
I/O register cancel (C → S)	No					No			
I/O Services									
data path resumed (S → C)	No					No			
data path suspended (S → C)	No					No			
fast data (C ←S)	No					No			
fast data (C → S)	No					No			
resume data path (C → S)	No					No			
send broadcast data (C → S)	No					No			
send data (S → C)	No					No			
send data (S ← C)	No					No			
send multicast data (C → S)	No					No			
start data path (S ← C)	No					No			
start data path (S ← C)	No					No			
stop data path (S → C)	No					No			
stop data path (S ← C)	No					No			
suspend data path (S → C)	No					No			
suspend data path (S ← C)	No					No			

Table 13 Supported CSTA Services and Events (Seite 8 von 11)

CSTA Services and Events	CSTA over TCP	Station, Group and Network Device Types				CSTA Over SIP	Client Device Types		ONS Support
		Normal	Type1	Hunt Group	Network Devices		OSMO	OpenStage (uaCSTA)	
DATA COLLECTION SERVICES	Not Supported								
Services									
Data Collected (S → C)	No					No			
Data Collection Resumed (S → C)	No					No			
Data Collection Suspended (S → C)	No					No			
Start Data Collection (C → S)	No					No			
Stop Data Collection (C ←→ S)	No					No			
Suspend Data Collection (C ←→ S)	No					No			
VOICE SERVICES & EVENTS	Not Supported								
Services (C → S)									
	No					No			
clear	No					No			
concatenate message	No					No			
deactivate	No					No			
delete message	No					No			
play message	No					No			
query voice attribute	No					No			
queue	No					No			
record message	No					No			
reposition	No					No			
resume	No					No			
review	No					No			
set voice attributes	No					No			
start	No					No			
stop	No					No			
suspend	No					No			
synthesize message (text to speech)	No					No			
Events									
bookmark reached	No					No			
completed	No					No			

Table 13 Supported CSTA Services and Events (Seite 9 von 11)

CSTA Services and Events	CSTA over TCP	Station, Group and Network Device Types				CSTA Over SIP	Client Device Types		ONS Support
		Normal	Type1	Hunt Group	Network Devices		OSMO	OpenStage (uaCSTA)	
DTMF detected	No					No			
emptied	No					No			
interrupted detection	No					No			
not recognized	No					No			
play	No					No			
recognized	No					No			
record	No					No			
review	No					No			
silence timeout expired	No					No			
started	No					No			
stop	No					No			
suspend play	No					No			
suspend record	**No**					No			
voice attribute changed	**No**					No			
voice error occurred	**No**					No			
CALL DETAIL RECORD (CDR) SERVICES	Not Supported								
Services									
call detail records notification (S → C)	No					No			
call detail records report (S → C)	No					No			
send stored call detail records (C → S)	No					No			
start call detail records transmission (S → C)	No					No			
stop call detail records transmission (C ←→ S)	No					No			
VENDOR SPECIFIC EXTENSIONS SERVICES & EVENTS	Not Supported								
Registration Services									
Escape Registration (C → S)	No					No			
Escape Registration Abort (S → C)	No					No			
Escape Registration Cancel (C → S)	No					No			
Services									
Escape (C ←→ S)	No					No			
Escape Services (Get Lower Class of Service)	No					No			

Table 13 Supported CSTA Services and Events (Seite 10 von 11)

CSTA Services and Events	CSTA over TCP	Station, Group and Network Device Types				CSTA Over SIP	Client Device Types		ONS Support
		Normal	Type1	Hunt Group	Network Devices		OSMO	OpenStage (uaCSTA)	
Escape Services (Set Lower Class of Service)						No			
Private Data Version Selection (C → S)	No					No			
Events									
Private Event	No					No			

Table 13 Supported CSTA Services and Events (Seite 11 von 11)

1 Get Configuration Data Response includes the subscriber's current message waiting status.
2 CSTA event monitoring is implicit and is started once the subscriber's CSTA over SIP client registers. Likewise monitoring is stopped once the CSTA over SIP client is no longer registered.
3 CSTA event monitoring is implicit and is started once the subscriber's CSTA over SIP client registers. Likewise monitoring is stopped once the CSTA over SIP client is no longer registered.
4 CSTA event monitoring is implicit and is started once the subscriber's CSTA over SIP client registers. Likewise monitoring is stopped once the CSTA over SIP client is no longer registered.
5 The Snapshot Device Response for OSMO includes a privateData tag that identifies the OSMO OND provisioned on OpenScape Voice V9.
6 Uses SIP 1st. party call control when OSMO-client is no longer registered.
7 Uses SIP 1st. party call control when OSMO-client is no longer registered.
8 Uses SIP 1st. party call control when OSMO-client is no longer registered.
9 Includes handover use cases using ONS

D Number Plan Considerations

OpenScape Voice V9 subscribers can be categorized in 5 basic types of subscribers:

- **Type I**: **Subscriber with public network number only**. This is the type of subscriber that is most used currently. This subscriber belong to a company that does not provide a private numbering plan.
- **Type II**: **Subscriber with private network number only**. This subscriber belong to a company that does provide a private numbering plan. This company probably does not have very many public network numbers and therefore, must create some users that do not have a public network number. The public network number presented for these users may be the BG Main Number or the number provisioned in the Outgoing Presentation Call Status feature – if assigned to the subscriber. [1]
- **Type III**: **Subscriber with private network number derived from public network number**. This subscriber has a public network number and a private network number linked using the extension part of the public number. For this subscriber, the office code of their Home DN uses the office code of the public network number.
- **Type IV**: **Subscriber with private network number not derived from public network number**. This subscriber has a public network number and a private network number where the extension part of the public network number and the extension part of the private network number are not the same. For this subscriber, the office code of their Home DN uses the office code of the public network number. The private network number cannot be displayed by OpenScape Voice V9. Only translation is capable of resolving the private network number to the subscriber who owns the private network number.
- **Type V**: **Subscriber that only has an extension**. These subscribers have neither a private network number nor a public network number. For this subscriber an office code with the leading three digits of the extension must be created. Another approach could be to force the definition of a private location code and force this type of subscriber into the Type II number scheme. OpenScape Voice V9 currently cannot configure this kind of subscriber.

Note: This Appendix provides information about the configuration of Type II numbers and explains how to properly configure these Type II numbers.

1. There are customers (for example IBM) that actually configure their switch with only Type II numbers and provide a separate public network number for each subscriber via the Outgoing Call Presentation Status feature

D.1 Configuring BGLs with Private Network Numbers Only

From an administrator's perspective, the administrator creates either a *private* or a *public* office code, then creates ranges of *private* or *public* Home DNs and finally assigns a free *private* or *public* home DN when creating a subscriber.

When introducing private numbering plans, the *External DN* flag must be unchecked when creating a subscriber with a Home DN that is considered 'private'.

This has as consequence that the office code and the Home Directory Numbers, created from the office codes, are actually private or public network agnostic and they should also be treated as such. Whether a Home DN is private or public actually only gets determined when the subscriber is created.

This understanding is very important for what follows – it explains why some actions which don't seem logical are actually necessary.

D.1.1 Office Code

When creating a *public* office code, the Country Code, Area Code and Local Office Code are filled in as determined by the public numbering plan authority.

There are 3 possible types of *private* office codes:

- **Level 2 Office Code** — Subscribers belong to 3 hierarchical levels: L0, L1 and L2. Each level is represented by a code.
- **Level 1 Office Code** —Subscribers belong to 2 hierarchical levels: L0 and L1. Each level is represented by a code.
- **Level 0 Office Code** —Subscribers have a single hierarchical level represented by their L0 code.

When creating a *private* office code, the digits of the L2 Code, L1 Code and L0 Code must be distributed amongst the Country Code, Area Code and Local Office Code fields of the office code. When OpenScape UC is involved, then it is important that all fields of the office code receive a value. The reason for this is that only office codes with country code, area code and local office code fields are synchronized from OpenScape Voice V9 to OpenScape UC and OpenScape UC does not allow the creation of office codes (it only allows the conversion of an office code to a private office code). Therefore, a private office code must be created according to the following table:

Private Office Code		**Office Code**		
		Country Code	**Area Code**	**Local Office Code**
Level 2		L2 Code	L1 Code	L0 Code
Level 1	L1 Code Length > 1 digit	First digit of L1 Code	Remainder of L1 Code	L0 Code
	L1 Code Length = 1 digit & L0 Code Length > 1 digit	L1 Code	First digit of L0 Code	Remainder of L0 Code
	L1 Code Length = 1 digit & L0 Code Length = 1 digit	**Not supported**		
Level 0	L0 Code Length >= 3 digits	First digit of L0 Code	Second digit of L0 Code	Remainder of L0 Code
	L0 Code Length < 3 digits	**Not supported**		

Examples:

Private Office Code		**Office Code**		
		Country Code	**Area Code**	**Local Office Code**
Level 2	58-57-56	58	57	56
Level 1	-57-56	5	7	56
	-5-756	5	7	56
	-5-7	**Not supported (at least 3 digits are required for the private office code)**		
Level 0	--567	**5**	6	7
	--56	**Not supported (at least 3 digits are required for the private office code)**		

As can be seen, the minimum private office code contains at least 3 digits and the digits are distributed among the available Office Code fields (Country Code, Area Code and Local Office Code).

D.1.2 Home Directory Numbers

Home Directory Numbers are created from the office code as before.

D.1.3 Subscribers

Subscribers must be created using the office codes and free Home Directory Numbers within the office code. For each subscriber to which a private Home DN is assigned the *External DN* must be unchecked at time of creation. Note that this flag cannot be modified once the subscriber is created.

D.1.4 Dial Plans and Routing

Administrators can use multiple ways in the numbering plan to route to one of these private subscribers.

- Direct Routing from the Destination Codes table
- Indirect Routing
  - Routing via Extensions table
  - Routing via Location Codes table

In the examples used in the following sections, a Level 1 Type II subscriber is created as follows:

Office code: 5758 (country code 5, area code 7, local office code 58)

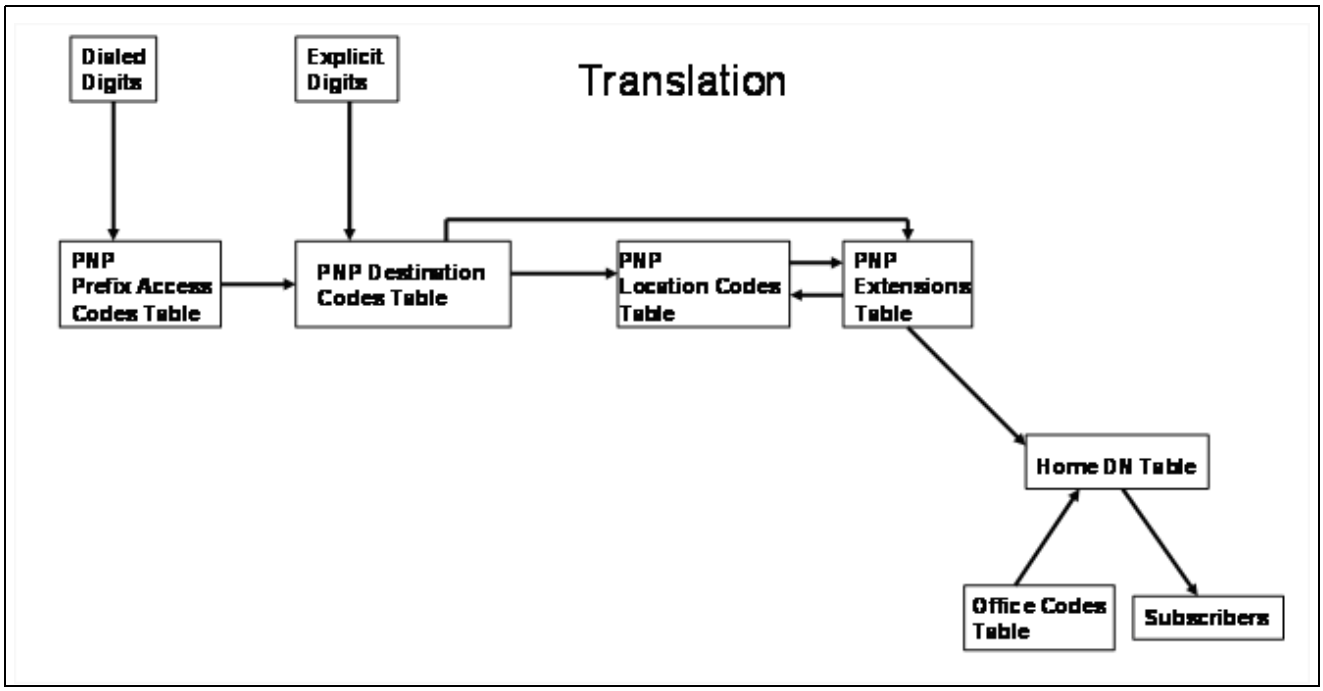
Home DNs: 57581001, 57581002

Subscribers: 57581001, 57581002

The L1Code is 57 and the L0 code is 58.

D.1.4.1 Direct Routing from Destinations Codes Table to Home DN

An administrator that wants to route calls for Type II subscribers directly to the Home DN table from the Destination Codes table must enter the destination codes table with the full Home DN and with the appropriate NOA (L2, L1 or L0).



For private numbers, the full Home DN – also known as the fully qualified private number – is required because OpenScape Voice cannot complete a lower level number to the highest level number because the office code that would be used to do these conversions does not necessarily contain the correct private level information. For the same reason, the appropriate NOA must be delivered to the Destination Codes table as well.

As an example of direct routing using the destinations codes table – if in a level 1 private numbering plan 2 subscribers of the same L1 code dial an L0 prefix access code and the L0 number of each other, the prefix access codes table should contain an entry for the L0 prefix code that completes the dialed number to the fully qualified L1 number by inserting the L1 code in the L0 PAC and setting the NOA to L1. The destination codes table can then be set up to point to the office code of the destination. The translation engine will then be able to output the correct private number level for the office code.

For example, suppose a Level 1 Type II subscriber (57581002) with office code 5-7-58. Suppose that level1 numbers do not need a prefix access code then, the entry in the PAC table to do extension dialing between the 2 subscribers would be something along the line of:

> PAC: 1; Min Length: 4; Max Length: 4; Digit Pos: 0; Insert: 5758; Prefix Type: On-net Access; Nature of Address: PNP Level 1; Destination Type: None; Destination Name: empty.

When 1002 is dialed, this PAC table rule will apply and because the Destination Type is none, the 57581002 number will be routed to the PNP Destination Codes table and try to find a rule there that matches the leading digits of the number and the nature of address: PNP Level 1. So, we must provide an entry in the PNP's destination codes table along the following line:

Dest Code: 5; Nature of Address: PNP Level 1; Destination Type: Home DN; DN Office Code: 5758.

The translation engine deliver as output the subscriber object that belongs to 57581002, the number and the nature of address that were presented to the Destination Codes table: 57581002/PNP Level 1.

D.1.4.2 Indirect Routing

If an administrator does not want to complete the number at the PAC level or if the explicit digits coming in from e.g. a SIP-Q gateway do not form a fully qualified private number for a Type II subscriber, then the only option is to go through the home extensions tables either using the home location table or directly from the destination codes table. Note that for this type of configuration, the use of the common numbering plan to resolve the translation is not allowed. In other words, it is not allowed to handle extensions in the common numbering plan, they must be handled in the private numbering plan of the subscriber requesting the translation.

Routing using Extensions Table

If the destination codes table contains an entry with NOA PNP Extension, then the administrator may route further translation to the Home Extensions table. Translation then tries to find a matching entry in the private numbering plan's extension table. If this is the extension of a subscriber with a private Home DN number then the extensions table is required to complete the number to the Home DN of the extension. To recognize that this is a private Home DN, a location code must be created with the exact same digits as the office code used to create the Home DN. This location code then breaks down the office code correctly into the private level codes. If the location code is not entered then the translation engine assumes that the office code is a public office code and returns an International NOA for the translation result.

As an example of indirect routing using the extensions table – if in a level 1 private numbering plan 2 subscribers of the same L0 code dial the extensions of each other, the prefix access codes table should contain an entry for the most significant digit of the extension, not modify the number and set the NOA to PNP Extension. The destination codes table can then be set up to point to the Home Extension table. A location code is then created with the exact same digits as the office code used to create the Home DN of the extension. The location code contains the correct breakdown of the private number allowing the translation engine to determine the level of the fully qualified private number. An entry is then created in the extensions table that points to the office code and the just created location code.

In the example, the Prefix Access Codes table would look as follows:

PAC: 1; Min Length: 4; Max Length: 4; Digit Pos: 0; Insert: none; Prefix Type: On-net Access; Nature of Address: PNP Extension; Destination Type: None; Destination Name: empty.

When 1002 is dialed, this PAC table rule applies and because the Destination Type is **None**, the 1002 number is routed to the PNP Destination Codes table and tries to find a rule there that matches the leading digits of the number and the nature of address: PNP Extension. So, an entry in the PNP's destination codes table must be provided along the following line:

Dest Code: 1; Nature of Address: PNP Extension; Destination Type: Home Extension.

Translation looks for an entry must be provided:

Prefix: 1, Length: 4, Location code: 5758, E164 Prefix: 57581, Destination Type: Home DN, Office Code: 5758.

Note that the Location Code 5758 should have been created already:

Code: 5758, L0 Length: 2, L1 Length: 2, L2 Length: 0, Number of digits to skip: 4

Because the location code and the office code are an exact match in the extension table, the translation engine delivers as output the subscriber object that belongs to 57581002, the completed number 57581002 and as nature of address the highest level read from the Location Codes table entry 5758 which is PNP Level 1 as the length of the L2 level is 0.

Routing using Location Codes tables

If the destination codes table contains an entry with NOA PNP Level 1 or PNP Level 0, then the administrator may route further translation to the Home Location table. The Location table is then used to strip the private number further down until only the extension remains. The extension is then passed to the Home Extension table and the same processing as for Routing using Extensions Table (Refer to Section D.1.4.2, "Indirect Routing") is done.

As an example of indirect routing via the location codes table – if in a level 1 private numbering plan 2 subscribers of the same L1 code dial an L0 prefix access code and the L0 number of each other, the prefix access codes table should contain an entry for the L0 prefix code, modify the number to remove the prefix and set the NOA to PNP Level 0. The destination codes table can then be set up to point to the Home Location table. A location code is then created with the exact same digits as the office code used to create the Home DN of the extension. An entry is then created in the extensions table that points to the office code and the just created location code, which will be used to determine the level of the fully qualified private number.

In the example, the Prefix Access Codes table would look as follows, assuming the prefix access code for L0 dialing is 7:

PAC: 7; Min Length: 7; Max Length: 7; Digit Pos: 1; Insert: none; Prefix Type: On-net Access; Nature of Address: PNP Level 0; Destination Type: None; Destination Name: empty.

When 7581002 is dialed, this PAC table rule applies and because the Destination Type is none, the 581002 number (the first digit is deleted) will be routed to the PNP Destination Codes table and try to find a rule there that matches the leading digits of the number and the nature of address: PNP Level 0. You must provide an entry in the PNP's destination codes table along the following line:

Dest Code: 5; Nature of Address: PNP Level 0; Destination Type: Home Location.

Translation now looks for an entry with matching leading L0 level digits in the PNP's Location codes table, so we must provide this entry:

Code: 5758, L0 Length: 2, L1 Length: 2, L2 Length: 0, Number of digits to skip: 4

Translation matches this as the L0 for this entry starts with 58. Translation skips the remaining 2 digits and present 1002 to the PNPs Extension table so we must provide this entry:

Prefix: 1, Length: 4, Location code: 5758, E164 Prefix: 57581, Destination Type: Home DN, Office Code: 5758.

Because the location code and the office codes are an exact match in the extension table, the translation engine deliver as output the subscriber object that belongs to 57581002, the completed number 57581002 and as nature of address the highest level read from the Location Codes table entry 5758 which is PNP Level 1 as the length of the L2 level is 0.

D.1.5 Number Modification

It may be necessary to enter information in the number modification tables to be able to understand and correctly break down numbers received from endpoints. Also for subscribers, the number definition tables need to be set up correctly in order for OpenScape Voice to understand how to break down a private number into its components. For OpenScape UC, Number Definition entries will be needed to add associated devices that are not known to OpenScape UC.

D.1.6 Restrictions

There are a few restrictions with this approach:

- Office codes of private Home DNs must have a minimum of 3 digits. This is because it must be possible to distribute these digits over the country code, area code and local office code.
- Office codes of private Home DNs are unique system-wide resources, because office codes themselves are unique resources. This means that 2 BG's cannot share the same private office codes or Home DNs.

When OpenScape UC is involved there are more restrictions:

- All subscribers with private numbers must have the same private number level (such as all of level 1 or all of level 2). If different levels are necessary then multiple OpenScape UC systems must be deployed.

- All private office codes must be made translatable in each PNP of OpenScape Voice (without the use of a barrier code and/or prefixes). Barrier codes and Level 2 or Level 1 prefixes are not allowed to be configured in the OpenScape UC configuration (Resources tab). OpenScape UC always gets the fully qualified private number from the OpenScape Voice CSTA service and may use this number on make call requests to OpenScape Voice. Therefore, OpenScape Voice must be ready to receive these numbers.

For example, if the dialing pattern for the OpenScape UC subscriber 57581002 is that no barrier code is required for on-net dialing and that a prefix 7 is required for dialing L0 numbers and that to dial L1 numbers a prefix 8 is required, then the OpenScape Client with number 57581002 is used to dial the extension of 57581001: 1001. To dial a number of a subscriber that is in the same L1 code (57), the OpenScape Client would be used to dial 7, followed by the L0 number of the destination subscriber. To dial a number of a subscriber that is in another L1 region, the OpenScape client would be used to dial 8, followed by the L1 number (fully qualified private number) of the destination subscriber. The office code 5-7-58 is synchronized to OpenScape UC and is converted to a Private Office Code:

PN Code ID: 5758, Barrier Code (must be empty), Level 2 Code: empty, Level 1 Code: 57, Level 0 Code: 58, Overlap: 0, Level 2 Prefix (must be empty), Level 1 Prefix (must be empty).

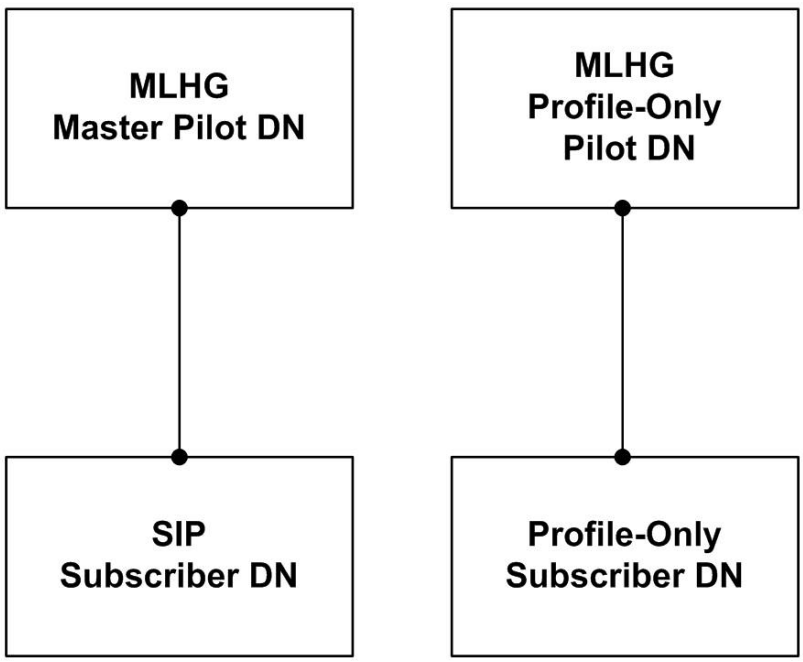
When the OpenScape client 1002 tries to add the extension 1001 as an associated device, OpenScape UC will pass 57581002 and 1001 to OpenScape Voice V9 which will look up the numbering plan used by 57581002, translate 1001 using this numbering plan, find that this is a Home DN (57581001, PNP Level 1) and it will deliver the fully qualified private number 57581001 back to OpenScape UC which will store this number for correlation and further usage. Whenever the OpenScape Client uses 1001 as an associated device, OpenScape UC creates a dialable number for 57581001 by adding the barrier code and the prefixes. As it is not allowed to enter values here, OpenScape UC will request OpenScape Voice V9 to use 57581001 for this call. Therefore, OpenScape Voice V9 needs to be able to translate the fully qualified private number 57581001.

- The barrier code and the L2/L1 prefix codes fields in the Private Office Code configuration for OpenScape UC must remain empty. This does not mean that the OpenScape Desktop Client is not allowed to use the barrier codes and prefixes when dialing or assigning preferred devices. The reason for not allowing barrier codes and level2 or level 1 prefixes to be entered is that there are situations where OpenScape UC adds the wrong barrier codes and prefixes to a number. Currently, OpenScape may not always retrieve the barrier code from the correct location and therefore it is nearly impossible for an administrator to know which entries are required in the PNPs of the OpenScape UC subscribers. The exact problem occurs when a Type III subscriber adds a Type II subscriber as an associated device. For the Type II subscriber, OpenScape Voice returns a fully qualified private number to OpenScape UC. OpenScape UC tries to make a dialable number when the OpenScape Client tries to make a call using this Type II subscriber as an associated device; however it only has the public office code settings of the

Type III subscriber and does not know how to make the private number dialable. To do this, OpenScape UC currently looks for the first available private office code and uses its settings. These settings are not always guaranteed to be the right settings; hence the requirement to keep the barrier code and level 2 or level 1 prefix fields empty and make the fully qualified private numbers dialable in all relevant PNPs (probably it's best to push the handling of these fully qualified numbers to the common numbering plan) and use Direct Routing from Destinations Codes Table to Home DN.

E MLHG and CSTA Capabilities

OpenScape Voice Hunt Group (a.k.a., Multi-line Hunt Group - MLHG) service enables calls to be routed to an available member within the group. Hunting occurs only when the called device is the Hunt Group pilot DN. The pilot DN may be a profile only subscriber DN or a SIP subscriber DN (pilot master).



Calls placed directly to a member of the group, with the exception of a pilot master will not hunt and are considered private calls. A private call makes the hunt group member unavailable for hunt group calls.

Hunt Group members must be subscribers within the same Business Group as the Hunt Group pilot DN.

When a call is placed to a hunt group pilot DN hunting begins according to the hunt type assigned to the group; linear (conditional or unconditional), circular, uniform call distribution (UCD) parallel or manual (or CTI controlled). Calls are distributed to the first available member according to the rules of the hunting type. If all members in the group are busy for hunted calls then depending on the hunt group provisioning the call may receive busy treatment or queued until an agent is available. Queued calls provide audible indication to the calling party, such as ringback or announcement. Announcements may include periodic queue position notification.

A Hunt Groups pilot DN provisioned for CSTA enables applications to monitor and distribution calls. CSTA supports call distribution to members or non-member (e.g., private network or public network destinations). Backup call distribution is provided by OpenScape Voice in the event that the hunt group controlling application fails or intentionally taken off-line.

Hunt Group pilot DN provisioned for One Number Service (ONS) is also possible but has not yet been fully tested with OpenScape UC.

There are several Hunt Group attributes that, with proper provisioning, enhance hunting, queuing, announcements, night service, member capabilities, and subscriber re-routing.

The following subsections define the currently supported hunt group attributes.

E.1 Basic Hunt Group Attributes

E.1.1 Hunt Group – Pilot DN

- To provision pilot DN with CSTA and ONS the pilot DN may be a profile only, or SIP subscriber DN.
- The DN may be public or internal.
- Multiple pilots DN are possible by provisioning several DN with Remote Call Forwarding (RCF) to a single pilot DN.

E.1.2 Hunting and Distribution types (Hunting Algorithms)

- Linear – Hunting begins from first to last member. Distribution is made to the first available member.
  - Linear Conditional: Calls are routed to the overflow destination only if the hunting passes the last group member and there is no queue, the queue is full or the call has been queued for too long. This is the traditional Linear hunting type.
  - Linear Unconditional: Calls are immediately routed to the overflow destination if the hunting passes the last group member. This behavior is preferable if the overflow destination is again a Hunt Group, e.g. on another switch (network-wide Hunt Group). If no overflow destination has been provisioned, then the Linear Conditional type will be used.
- Circular – Hunting begins from member after last member to receive call and continues in circular fashion. Distribution is made to the first available member.
- Uniform Call Distribution (UCD) – Hunting begins at the member that has been idle the longest and continues to the next agent that has been idle longer than all others and so on. Distribution is made to the first available member.

- Manual (CSTA Application Controlled) – Inbound calls are immediately queued waiting for the application to distribution calls to agents or destinations. Also see Backup Call Distribution.
- Parallel – Call Pick Model: The distributed call alerts all available Hunt Group members in parallel. First available member received call, all other alert via special call pickup group arrangement.
- Parallel – Simultaneous Alerting Model: Available Hunt Group members are alerted via call setup simultaneously.

Note: Hunt Group Calls may NOT be deflected with binding using hunt group pilot as the ONS.
Hunt Groups may be used as preferred devices (OND) to hold calls, for example IBM “Ask Me” queue.

Attention: One Number Service (ONS) is internal to Unify. Refer to Appendix F, “One Number Service (ONS)” for details.

E.1.3 Queuing Attributes

E.1.3.1 Maximum Queue size

This defines the maximum number of calls that may be queued to the group depending on the configured queue size.

E.1.3.2 Configured Queue size

Configured queue size must be greater than or equal to maximum queue size. This defines the actual number of calls that may be queued to the group.

E.1.3.3 Queue Priority

Value (1 to 255) determines the queue search order when an agent is a member of multiple groups. The lower the number the higher the priority.

E.1.3.4 Overflow Destination

Defines where call should be moved if the Queue is full or the call has spent the maximum configured time in queue. For example, this could be group voice mail.

E.1.3.5 Maximum Time In Queue

Defines the maximum time a call remains in queue before being moved to the Overflow Destination.

E.1.3.6 Intercept Announcement

Defines the intercept treatment provided to the calling device when the call is queued.

E.1.3.7 Queue Position Announcement Interval and Queue Position Announcement

The interval defines when a queued call is moved to the Queue Position Announcement and then returned to the intercept announcement.

E.1.3.8 Maximum Time In Queue

Defines the maximum time a call remains in queue before being moved to the Overflow Destination.

E.1.4 Hunting Attributes

E.1.4.1 No Answer Advance

Defines the duration that a distributed call will alert the selected member until the call is returned to the group to be hunting to another available member.

The auto make busy attribute may be used to cause the member to be made busy for subsequent hunt group calls.

Note: This attribute will have an impact on the UC application's rules and device list processing. Setting the timer too low may cause the hunt group process to divert the call back to the group and re-hunt. This will cause the UC application to loose control of the call much like phone call forwarding.
The No Answer Advance attribute is suspended for calls that are distributed by the UC application (manual mode).

E.1.4.2 Auto Make Busy

If No Answer Advance timer expires this attribute determines if the alerted member should automatically be made busy.

The member remains busy for hunt group calls until manually changed.

E.1.4.3 Post Call Time (e.g., Wrap-up)

Defines how long after a member has answered a call does the member remain temporarily not available for a hunted call. This attribute affects agent "working after call" state.

Note: This option is not used applied to calls that are distributed by the UC application. Application may set "working after call" via Agent State services.

E.1.4.4 Collect Traffic Measurement Data

Activates the collection of CDR traffic data for the group.

E.1.5 Night Service Attributes

E.1.5.1 Enable Night Service

Activates or deactivates Night Service for the group. When active calls are not hunted but instead are moved to either the provisioned Night Service Directory Number or Overflow Destination. If neither is provisioned then busy treatment is applied.

E.1.5.2 Night Directory Number

Destination where calls are routed if Night service is enable. This number may be internal or off-network.

E.1.5.3 Attendant

If set this option directs all incoming call to the system attendant number.

E.1.6 Member / Agent Attributes

E.1.6.1 Subscriber ID

Defines the OpenScape Voice subscriber that is a member of the hunt group.

E.1.6.2 Position

Defines the position in the hunt group that a member is assigned. This determines the hunting order in a linear hunting arrangement.

E.1.6.3 Queue Priority

Defines member's queue priority used if member is provisioned in multiple hunt groups.

E.1.6.4 Busy Status

Sets and/or indicates the Hunt Make Busy Status of the member. Busy agents are not hunted.

Note: Busy status may be set via CSTA Set Agent – Ready or Not Ready. The CSTA Get Agent State service is also supported. A CSTA interface option enables this status for specific hunt groups where the agent is a member.

E.1.6.5 Busy Stop Hunt

Sets and/or indicates the Stop Hunt status of the member. If active hunting stops at this member.

E.1.6.6 Can Make Hunt Group Busy

Activate or deactivates permission for a member to activate the Hunt Make Busy feature.

E.1.6.7 Can Stop Hunt Group Hunting

Activate or deactivates permission for a member to activate the Stop Hunt feature.

E.1.6.8 Attendant

Enables or disables CSTA agent state event flow for this member.

E.1.7 Call Admission Control and WAN Failure

E.1.7.1 Enable Enhanced Subscriber Rerouting

This parameter controls whether Enhanced Subscriber Rerouting is allowed in case a distributed call to a Hunt Group member is rejected due to Call Admission Control.

Note: It is NOT possible to set this parameter for Hunt Groups with Hunting Type set to Parallel.

E.1.7.2 Enable Basic Subscriber Rerouting

This parameter controls whether Basic Subscriber Rerouting is allowed in case a distributed call to a Hunt Group member is rejected due to Call Admission Control or WAN Failure.

It is NOT possible to set this parameter for Hunt Groups with Hunting Type set to either Manual or Parallel.

Note: This parameter replaced the former “Enable Rerouting” parameter

---

E.1.8 Calling Party Display

Note: In a future release the administrator will have the option to configure a subscriber that is a MLHG member to use the Name and Number of the Main Pilot DN as the subscriber’s identity for Internal and/or External calls.

For calls distributed from the hunt group, if the calling party is an internal subscriber the number of the hunt group pilot number will be displayed on the phone and provided via CSTA events.

For outbound calls from an agent the Main Pilot DN (first MLHG in agents list) will be presented as the user’s identity. If the called party is external and the Main Pilot DN is internal then the BG Main Number is used.

E.1.9 Manual (Application Controlled) Hunting and Backup Call Distribution (BCD)

Backup Call Distribution (BCD) requires that the application support Application Session Services and the private extension for applicationHeartbeat timer.

BCD applies to a Hunt Group provisioned for Manual (Application Controlled) hunting type.

BCD is active while no CSTA monitor has been started on the hunt group's pilot DN.

When BCD is active all inbound calls will be automatically distributed flowing UCD hunting method.

Once a CSTA monitor has been started on the hunt group's pilot DN BCD is deactivated and calls are queued waiting for the application to distribute calls to member or elsewhere.

F One Number Service (ONS)

One Number Service (ONS) is an extension of the OpenScape Voice V9 CSTA interface. It is Unify specific, and is used exclusively between OpenScape Voice V9 and OpenScape UC Application to create, route, track and, control inbound and outbound calls via standard CSTA call control services for any device located anywhere.

When ONS connections are created, the “device identification” is extended to include a One Number Service Device (OND) tag. ONS is the logical OpenScape Voice V9 subscriber identity used to provide call control services. OND is the directory number of the physical device used for the media; this may be another OSV subscriber device or external (off-net) device.

Within the CSTA event flow, device IDs may be presented on device monitors with an “ond” tag.

Note: Third Party Applications must accept the deviceID string with the “ond” tag. This conforms to the ECMA 269 CSTA connection model. The “ond’ tag would not be used and must be ignored by Third Party Applications.

The following xml example illustrates the established connection parameter presentation in an ONS call:

```
<establishedConnection>
    <callID>FF0002000000000006672494DE6040000</callID>
    <deviceID>N&lt;+15557654321&gt;;ond=+15551234567</deviceID>
</establishedConnection>
```

Note: In a connection identifier, the device Name is always presented as an empty string. The “ond” tag begins after the semi-colon (;) in the NM field.

When an ONS connection is cleared, then the Clear Connection service shall include the “ond” tag in the connectionToBeCleared parameter, as illustrated in the xml example below:

```
<?xml version="1.0" encoding="utf-8"?>
<ClearConnection xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.ecma-
international.org/standards/ecma-323/csta/ed3">
    <connectionToBeCleared>
        <deviceID> N&lt;+ 15557654321&gt;;ond=+15551234567</deviceID>
        <callID> FF0002000000000006672494DE6040000</callID>
    </connectionToBeCleared>
</ClearConnection>
```

List of Abbreviations

This table shows some important abbreviations.

Abbreviation	Definition
ACD	Automatic Call Distribution; Telephone facility that manages incoming calls and handles them based on the number called and an associated database of handling instruACDs to validate callers make outgoing responses or calls, forward calls to the right party, allow callers to record messages, gather usage statistics, balance the use of phone lines, and provide other services. Many companies offering sales and service support use
AICS	Automatic Incoming Call Selection; called Call Forcing in this document
AIN	Advanced Intelligent Network
API	Application Programming Interface
B50	SEPP milestone at which a feature is defined
BC	Business Connect
BG	Business Group; also called Centrex Group or Tenant Group
BL	Busy Line
BO	Business Opportunity
CAC	Call Admission Control
CALEA	Communications Assistance for Law Enforcement Act (J-STD-025); also known as Lawful Intercept
CAP	Common Application Platform, host platform for Com Assistant
CDR	Call Detail Record
CLI	Command Line Interface
CPE	Customer Premises Equipment
CSTA	Computer Supported Telecommunications Applications; ECMA open-standard interface definition to enable computing systems to control telephony systems (3rd party call control)
CTI	Computer Telephony Integration the interface that enables first or third party application control of voice features.
DB	Database
DD	Dialable Digits
DLS	Download Server for optiPoint endpoints
DN	Directory Number
DNIS	Dialed Number Identification Service
DSS	Direct Station Selection
ECMA	European Computer Manufacturers Association; Maintainers of the CSTA standards
FRN	Feature Request Number (replaces LM Number)

Abbreviation	Definition
FRU	Field Replaceable Unit (a.k.a. Spare)
FW	Firmware
GNF	Global Number Format
GUI	Graphical User Interface
HMB	Hunt Make Busy
OpenScape Voice	Collective name Unify soft switch products. Unless otherwise noted this applies to OpenScape Voice cluster.
HW	Hardware
IKE	Internet Key Exchange
IM	Instant Message
IP	Internet Protocol
IPSec	Internet Protocol Security
IUS	State of an inspected, released document
LAN	Local Area Network
LM	Leistungsmerkmal (German for “Feature”); SNC uses the term FRN instead
Make-Busy	Name only used internally; the OpenScape Voice name for this ACD agent state is Unavailable
MD	Manufacture Discontinue; End of Life for a product
MGCP	Media Gateway Control Protocol; IETF standard RFC 2705
MTA	Multimedia Terminal Adaptor (for example, Packet Cable or MGCP)
MLHG	Multiline Hunt Group
NAA	No Answer Advance (an MLHG feature)
NMC	Network Management Center
OEM	Original Equipment Manufacturer
OND	One Number Service Device
ONS	One Number Service
ONS-IO	One Number Service – Inbound Outbound
OS	Operating System
PC	Personal Computer
PLM	Product Line Management
scx	Unify CSTA Extension – for example, used in privateData and other parameters
SE	Systems Engineering
SFR	Switching Function Representation
SIP	Session Initiation Protocol; an IP-based protocol for distributed applications; IETF standard RFC-3261
SIP IAD	SIP Integrated Access Devices for analog support.

Abbreviation	Definition
SMU	Staff Month Unloaded
SNMP	Simple Network Management Protocol; IETF standard
SOAP	Simple Object Access Protocol; IETF standard RFC 3288
SS7	Signaling System Number 7
SST	Single Step Transfer
SW	Software
TCP	Transmission Control Protocol
TLS	Transparent LAN Service
TTUD	TLS/TCP UDP Dispatcher
uaCSTA	User Agent CSTA Services over SIP transport
UCD	Universal Call Distribution
UI	User Interface
Work	ACD agent state; work state is used by an agent to complete the paperwork associated with a call that has just completed; sometimes referred to as Post-Call or Wrap-up time.
Wrap-up	Name not used; the OpenScape Voice name for this ACD agent state is Work
XML	Extensible Markup Language

The CSTA definitions and abbreviations are defined in ECMA TR/72

Index

A
accept call 109
administration 55
agent busy event 239
agent not ready event 240
agent ready event 242
agent working after call event 243
alternate call 110
answer call 111
application connectivity 55
application heartbeat 104
application link security considerations 29
application session services 100
application session terminated 108
application white list 29
associate data service 189

B
back in service event 251
basic application session 103
BGL services 60
bridged event 144

C
call admission control (CAC) subscriber re-routing 67
call associated features 189
call back call-related 113
call back event 245
call back non-call related 226
call cleared event 144
call control 109
call information event 194
call scenarios 270
camp on call 115
cancel callback service 227
cancel telephone tones service 190
change connection information service 191
change system status filter 81
clear call 115
clear connection 116
cold stand-by operation 66
compliance statement 21
conference call 118
conferenced event 144
connection cleared event 147
consultation call 120
CSTA Application Connectivity Example 22
CSTA services and events 272

D
data collection services 255
deflect call 124
delivered event 150
device capabilities changed event 253
device maintenance events 250
DeviceID 32
dial digits 125
dialable digits 33
dialed digits event 153
digits generated event 197
directed pickup call 125
diverted event 154
do not disturb event 246
duplex OpenScape Voice system 66
dynamic feature presentation 47

E
ECMA 354 application session services 67
ECMA-323 namespace 21
established event 157
event causes 155
events 194
events by device type 272
Example
    OpenScape Voice Hunt Group and OpenScape UC Application 62

F
failed event 160
failure and restart considerations 66
fallback call distribution 104
forwarding event 248

G
generate digits service 192
generate telephony tones service 193
get agent state service 227
get CSTA features 69
get do not disturb service 231
get forwarding service 235
get logical device information 70
get message waiting indicator service 218
get microphone mute service 219
get speaker volume service 222
get switching function capabilities 74

global number format (GNF) 38
global number format (GNF) replacement -- RTP parameter 65
GNFPrefixReplacement 65
group pickup call 125

H
held event 164
hold call service 126
hot stand by operation 66

I
I/O services 255
intrude call 127
IPSec 29

J
join call 127

K
keyset operation 64

L
logical device feature events 239
logical device feature services 225

M
make call 130
make predictive call 134
management 55
message waiting event 225
MLHG and CSTA Capabilities 294
MLHG pilot (group device/ACD) and agent BGLs 62
monitor recovery 103
monitor start 90
monitor stop 93
monitoring service 89
multibyte support for name (NM) field 43
multiple appearance 64

N
negative service response 43
network capabilities changed event 165
network reached event 166
non-registering SIP device 60
number and name privacy 39

O
offered event 169
OpenScape Voice privateData schema 21, 271
operational model 21
originated event 174
out of service event 254

P
park call 135
physical device events 225
physical device features 217

Q
queued event 177

R
reconnect call 136
registering SIP device 60
registration services 81
reset application session timer 107
retrieved event 181
routing services 203

S
send message 137
send user information service 193
service initiated 183
service request parameters 139
services by device type 272
set agent state 229
set do not disturb service 233
set forwarding service 237
set microphone mute service 223
set speaker volume service 224
SFR fields 34
SFR tags 36
simplex OpenScape Voice system 66
single step conference call 138
single step transfer call 139
SIP multiple contact registration 63
SIP phone/client configuration data 61
snapshot call 94
snapshot device 96
snapshot services 94
start application session 101
subscriber management configuration 60
supported CSTA services and events 30, 272
supported services and events 30
switching function representation (SFR) 33
system services 81

T
telephony tones generated event 197
transfer call events 144
transfer call service 142
transferred event 186
transport considerations 23
transport security 29
typical subscriber services 60

U

uaCSTA environment 25

V

vendor specific externsions service 255
voice services and events 255