

Cisco BroadWorks

Computer Telephony Integration

Interface Specification

Release 23.0
Document Version 4



Notification

The BroadSoft BroadWorks has been renamed to Cisco BroadWorks. Beginning in September 2018, you will begin to see the Cisco name and company logo, along with the new product name on the software, documentation, and packaging. During this transition process, you may see both BroadSoft and Cisco brands and former product names. These products meet the same high standards and quality that both BroadSoft and Cisco are known for in the industry.

Copyright Notice

Copyright[©] 2019 Cisco BroadWorks, Inc. All rights reserved.

Trademarks

Any product names mentioned in this document may be trademarks or registered trademarks of Cisco or their respective companies and are hereby acknowledged.



Document Revision History

Release	Version	Reason for Change	Date	Author
17.0	0.1	Created initial draft.	September 22, 2009	Stephan Goulet
17.0	0.1.2	New template proposal.	September 28, 2009	Stephan Goulet
17.0	0.1.3	Added code definitions, Call Control actions, events, and call flows.	October 2, 2009	Stephan Goulet
17.0	0.1.4	Updated call management and conference events. Added placeholder for all missing actions. Added consultative transfer information.	October 9, 2009	Stephan Goulet
17.0	0.1.5	Added placeholders for missing elements. Added agent state information.	October 14, 2009	Stephan Goulet
17.0	0.2	Updated based on comments from 0.1.5 review.	October 15, 2009	Stephan Goulet
17.0	0.2.1	Added conference actions.	October 19, 2009	John Seeto
17.0	0.2.2	Added Route Point and Automatic Call Distribution (ACD) concepts, actions, events, and call flows.	November 04, 2009	Stephan Goulet
17.0	0.3	Updated based on comments from internal review.	November 06, 2009	Stephan Goulet
17.0	0.3.1	Added missing call management actions and agent actions.	November 16, 2009	Stephan Goulet
17.0	0.4	Updated document based on comments from internal review.	November 20, 2009	Stephan Goulet
17.0	0.5	Added system and call concept section. Added <i>Appendix A Changed Event Types</i> .	December 2, 2009	Martin Perron Bob Russell Stephan Goulet
17.0	0.5.1	Improved and added missing messages flows.	December 4, 2009	Eric Bernier
17.0	1.0	Updated based on comments from internal review.	December 21, 2009	Eric Bernier Martin Perron Omar Paul Stephan Goulet
17.0	1.1	Updated based on comments from external review.	March 20, 2010	Stephan Goulet
17.0	1.2	Added automated call flows and XML schema descriptions.	April 30, 2010	Martin Perron Stephan Goulet
17.0	1.3	Made minor editorial changes.	April 30, 2010	Patricia Renaud
17.0	1.4	Updated section 6.2.2 Call Originated for EV 111837, section 6.2.5 Call Held for EV 111900, and section 6.2.6 Call Retrieved for EV 111902.	May 26, 2010	Goska Auerbach



Release	Version	Reason for Change	Date	Author
17.0	1.4	Updated event package name, <i>Monitor Next Call</i> request description and various call flows based on the latest implementation.	May 27, 2010	Stephan Goulet
17.0	1.4	Updated section 5 Requests for EV 112516 and section 6.5 ACD Events for EV 112367. Clarified <i>Route Point</i> event completion in section 4.3.6 Media Playback and Digits Collection .	June 2, 2010	Stephan Goulet
17.0	1.4	Updated section 3.4.5.5 Target Scope Rules for EV 112654. Edited and published document.	June 18, 2010	Margot Hovey-Ritter
17.0	1.5	Made minor editorial changes.	August 30, 2010	Jessica Boyle
17.0	1.5	Added error codes. Updated section 5 Requests with new error codes for EV 113350.	August 31, 2010	Stephan Goulet
17.0	1.5	Edited changes.	September 9, 2010	Patricia Renaud
17.0	1.5	Edited and published document.	September 14, 2010	Patricia Renaud
17.0	1.6	Updated section 5 Requests for EVs 117895 and 117612.	September 15, 2010	Stephan Goulet
17.0	1.6	Edited changes and published document.	September 17, 2010	Andrea Fitzwilliam
17.0	1.7	Updated section 3 System Concepts , 4 Call and Service Management Concepts , and 5 Requests for EV 120600.	January 4, 2011	Stephan Goulet
17.0	1.8	Updated section 5.2.15 Monitor Call for EV 111969.	January 26, 2011	Stephan Goulet
17.0	1.9	Updated section 5 Requests for EV 121906.	January 26, 2011	Stephan Goulet
17.0	1.10	Updated section 6.3.1 Conference Started for EV 130182. Added section 10 Error Code List for EV 120600.	April 8, 2011	Stephan Goulet
17.0	1.11	Updated section 5.12 Application Controller Request for EV 117612.	October 7, 2011	Stephan Goulet
17.0	1.12	Introduced changes for Releases 17.sp2, 17.sp3, and 17.sp4.	November 21, 2011	Martin Perron
18.0	1	Updated document for Release 18.0.	November 21, 2011	Martin Perron
18.0	1	Edited changes and published document.	January 20, 2012	Patricia Renaud
18.0	2	Updated sections 6.1 Event and Event Response and 6.15.1 Channel Terminated for EV 146293.	May 4, 2012	Martin Perron
18.0	2	Edited changes and published document.	June 11, 2012	Andrea Fitzwilliam
19.0	1	Updated document for EV 159752, EV 159750, EV 168327, and EV 131756.	August 23, 2012	Stephan Goulet



Release	Version	Reason for Change	Date	Author
19.0	1	Updated section 5.9.7 Modify Hoteling Guest for EV 167024.	August 30, 2012	Stephan Goulet
19.0	1	Added Meet-Me Conferencing functionality.	September 17, 2012	Stephan Goulet
19.0	1	Introduced changes for Release 19.0.	September 25, 2012	Stephan Goulet
19.0	1	Updated section 5.6 Agent Requests (AS only) for EV 174309.	October 9, 2012	Stephan Goulet
19.0	1	Updated keepalive description in section 3.4 Events .	October 17, 2012	Stephan Goulet
19.0	1	Edited changes and published document.	December 11, 2012	Patricia Renaud
20.0	1	Introduced changes for Release 20.0.	October 3, 2013	Stephan Goulet
20.0	1	Updated section 3.6 Backward Compatibility for EV 185622.	October 11, 2013	Stephan Goulet
20.0	1	Updated section 6.10 Call Park Events for EV 193880.	October 11, 2013	Stephan Goulet
20.0	1	Updated section 5.7.2 Outgoing Dial for EV 197892.	October 11, 2013	Stephan Goulet
20.0	1	Added section 7.5 Character Sets Encoding for EV 159081.	October 29, 2013	Stephan Goulet
20.0	1	Updated section 6.11 Meet-Me Conference Events (AS only) for EV 200958.	November 3, 2013	Stephan Goulet
20.0	1	Updated section 6.2 Call Events for EV 187391.	November 20, 2013	Stephan Goulet
20.0	1	Edited changes and published document.	December 3, 2013	Joan Renaud
20.0	2	Added section 5.4.3 Play Treatment that was missing from the document for EV 218655.	March 13, 2014	Goska Auerbach
20.0	2	Edited changes and published document.	July 28, 2014	Joan Renaud
21.0	1	Introduced changes for Release 21.0.	November 17, 2014	Martin Perron
21.0	1	Updated the legal notice. Edited changes.	November 19, 2014	Joan Renaud
21.0	1	Rebranded and published document.	November 20, 2014	Joan Renaud
21.0	2	Updated server icons and published document.	February 26, 2015	Joan Renaud
21.0	3	Removed maxChannelsReached reason from channelTerminatedEvent in section 8.2.3 ChannelTerminatedReason for PR-47018.	April 9, 2015	Alexandre Beaudry
21.0	3	Clarified system domain usage for the Get Agent State request in section 5.6.1.1 Description for PR-45563.	April 20, 2015	Martin Perron
21.0	3	Edited changes and published document. Verified <i>Table of Figures</i> .	June 7, 2015	Jessica Boyle
22.0	1	Introduced changes for Release 22.0.	September 21, 2016	Sébastien Martineau



Release	Version	Reason for Change	Date	Author
22.0	1	Edited changes.	November 29, 2016	Joan Renaud
22.0	1	Added section 11 Acronyms and Abbreviations . Edited changes and published document.	December 14, 2016	Joan Renaud
23.0	1	Introduced changes for Release 23.0.	June 27, 2018	Sébastien Martineau
23.0	1	Rebranded document for Cisco. Edited changes and published document.	September 28, 2018	Joan Renaud
23.0	2	Edited section 6.2.3 Call Received for PR-59835.	October 29, 2018	Sébastien Martineau
23.0	2	Edited changes and published document.	February 6, 2019	Jessica Boyle
23.0	3	Rebranded product name for Cisco and published document.	March 19, 2019	Joan Renaud
23.0	4	Added section 7.2.1 URI Encoding for PR-58772.	April 22, 2019	Sébastien Martineau
23.0	4	Edited changes and published document.	August 13, 2019	Jessica Boyle

Table of Contents

1 Summary of Changes	25
1.1 Changes for Release 23.0, Document Version 4	25
1.2 Changes for Release 23.0, Document Version 3	25
1.3 Changes for Release 23.0, Document Version 2	25
1.4 Changes for Release 22.0, Document Version 1	25
1.5 Changes for Release 21.0, Document Version 3	25
1.6 Changes for Release 21.0, Document Version 2	25
1.7 Changes for Release 21.0, Document Version 1	25
1.8 Changes for Release 20.0, Document Version 2	26
1.9 Changes for Release 20.0, Document Version 1	26
1.10 Changes for Release 19.0, Document Version 1	26
1.11 Changes for Release 18.0, Document Version 2	27
1.12 Changes for Release 18.0, Document Version 1	27
1.13 Changes for Release 17.sp4, Document Version 1.12	27
1.14 Changes for Release 17.sp3, Document Version 1.12	27
1.15 Changes for Releases 17.sp2, Document Version 1.12	28
1.16 Changes for Release 17.0, Document Version 1.11	28
1.17 Changes for Release 17.0, Document Version 1.10	28
1.18 Changes for Release 17.0, Document Version 1.9	28
1.19 Changes for Release 17.0, Document Version 1.8	28
1.20 Changes for Release 17.0, Document Version 1.7	28
1.21 Changes for Release 17.0, Document Version 1.6	28
1.22 Changes for Release 17.0, Document Version 1.5	28
1.23 Changes for Release 17.0, Document Version 1.4	29
1.24 Changes for Release 17.0, Document Version 1.3	29
1.25 Changes for Release 17.0, Document Version 1.2	29
1.26 Changes for Release 17.0, Document Version 1.1	29
2 Overview.....	30
2.1 Purpose	30
2.1.1 Document Structure.....	30
2.2 Computer Telephony Integration Interface.....	31
2.2.1 Overview	31
2.2.2 Cisco BroadWorks Application.....	31
2.2.3 Functionality	31
2.2.4 CTI Messaging.....	32
2.2.5 Cisco BroadWorks Deployment Mode	32
3 System Concepts.....	34
3.1 Overview	34
3.2 High-Availability Solution	35

3.3 Requests	35
3.3.1 Authentication Scheme and Session Management.....	36
3.3.2 Trusted Client Application.....	36
3.4 Events.....	36
3.4.1 High-Availability Configuration One – Active-Standby with Synchronization	38
3.4.2 High-Availability Configuration Two – Active-Active with Synchronization	39
3.4.3 High-Availability Configuration Three – Active-Active Without Synchronization	39
3.4.4 Event Channel	40
3.4.5 Event Subscription	42
3.4.6 Application Controller (AS only).....	50
3.5 Initialization.....	52
3.6 Backward Compatibility	53
3.6.1 Requests	53
3.6.2 Events.....	54
3.6.3 Version Value.....	54
3.6.4 XML Schema	54
4 Call and Service Management Concepts	56
4.1 Call Model	56
4.1.1 Subscriber	56
4.1.2 Subscriber ID	56
4.1.3 User ID	56
4.1.4 Address	56
4.1.5 Call Session	57
4.1.6 Call.....	58
4.1.7 Endpoint	67
4.1.8 Party	68
4.1.9 Topologies.....	68
4.1.10 Redirection	71
4.1.11 Executive Assistant (AS only)	71
4.2 Conference Model	72
4.2.1 Conference States	72
4.2.2 Conference Requests.....	73
4.2.3 Conference Events	73
4.2.4 Conference State Table	74
4.2.5 Minimum Number of Participants in Conference	74
4.3 Route Point (AS only)	75
4.3.1 Incoming Call Processing.....	75
4.3.2 Route Point Requests.....	78
4.3.3 Route Point Events	78
4.3.4 DNIS	80
4.3.5 Route Point Failure Policy	80
4.3.6 Media Playback and Digits Collection	81

4.4	ACD (AS only).....	83
4.4.1	ACD Types.....	83
4.4.2	Incoming Call Processing.....	84
4.4.3	ACD Requests	86
4.4.4	ACD Events.....	87
4.4.5	Agent	88
4.4.6	Supervisor	90
4.4.7	Threshold and Alerts.....	90
4.5	Call Park	91
4.6	Meet-Me Conference (AS only)	91
4.6.1	Meet-Me Conference Types	91
4.6.2	Roles.....	91
4.6.3	Conference Start.....	92
4.6.4	Conference Join.....	92
4.6.5	Capabilities	93
4.6.6	Meet-Me Conferencing Request.....	94
4.6.7	Meet-Me Conferencing Events	94
4.7	Collaborate Room (AS only)	95
4.7.1	Collaborate Room Types	96
4.7.2	Roles.....	96
4.7.3	Collaborate Room Start.....	96
4.7.4	Collaborate Room Join.....	97
4.7.5	Capabilities	97
4.7.6	Collaborate Room Request.....	97
4.7.7	Collaborate Room Events	98
4.8	Service Management	99
4.8.1	Service Description.....	99
4.8.2	Service Management Requests.....	103
4.8.3	Service Management Events	103
4.9	Voice Messaging (AS only).....	105
4.9.1	Voice Messaging Request	105
5	Requests	106
5.1	Requests and Responses.....	106
5.1.1	Request	106
5.1.2	Response	106
5.1.3	Common Errors	107
5.2	Call Management Requests.....	110
5.2.1	Dial.....	110
5.2.2	Talk.....	117
5.2.3	Hold	123
5.2.4	Release	126
5.2.5	Blind Transfer.....	130

5.2.6	Consultative Transfer.....	133
5.2.7	Mute Transfer.....	137
5.2.8	Reconnect.....	140
5.2.9	Transfer to Voice Mail.....	142
5.2.10	Transmit DTMF	144
5.2.11	Call Return	146
5.2.12	Customer Originated Trace.....	149
5.2.13	Directed Call Pickup	150
5.2.14	Directed Call Pickup with Barge-in.....	155
5.2.15	Monitor Call	161
5.2.16	Monitor Next Call (AS only).....	168
5.2.17	Last Number Redial.....	178
5.2.18	Retrieve Call List.....	181
5.2.19	Retrieve Call Details	181
5.2.20	Call Me Now (AS only)	182
5.2.21	Record (AS only).....	184
5.2.22	Retrieve	186
5.2.23	Find-me/Follow-me Call Push (AS only)	189
5.2.24	Allocate IMRN	192
5.2.25	Executive-Assistant Initiate Call (AS only).....	193
5.2.26	Executive-Assistant Call Push (AS only)	196
5.2.27	Pause Recording (AS only).....	198
5.2.28	Resume Recording (AS only)	199
5.2.29	Stop Recording (AS only).....	200
5.2.30	Set Client Session Info (AS only).....	201
5.2.31	Retrieve Hook Status (AS only).....	202
5.3	Conference Requests.....	203
5.3.1	Conference Start.....	203
5.3.2	Conference Release.....	206
5.3.3	Conference Hold	207
5.3.4	Conference Retrieve.....	209
5.3.5	Conference Mute	210
5.3.6	Conference Unmute	211
5.3.7	Conference Add Participant	213
5.3.8	Mute Call	214
5.3.9	Unmute Call	215
5.3.10	Deaf Call.....	217
5.3.11	Undeaf Call.....	218
5.4	Route Point Requests (AS only)	219
5.4.1	Get Route Point	219
5.4.2	Distribute Call	220
5.4.3	Play Treatment.....	223

5.4.4	Play Music On Hold	224
5.4.5	Play Ringback	226
5.4.6	Play Silence.....	227
5.4.7	Play Busy	229
5.4.8	Outgoing Dial	230
5.4.9	Bounce Call.....	234
5.4.10	Blind Transfer.....	237
5.4.11	Release Call.....	238
5.4.12	Fail Call.....	240
5.4.13	Get Route Point State.....	242
5.4.14	Modify Route Point State.....	243
5.5	ACD Requests (AS only).....	245
5.5.1	Get ACD	245
5.5.2	Blind Transfer.....	246
5.5.3	Promote Call	247
5.5.4	Reorder Call	249
5.5.5	Release Call.....	250
5.6	Agent Requests (AS only).....	252
5.6.1	Get Agent State	252
5.6.2	Set Agent State.....	253
5.6.3	Tag Disposition Code	253
5.6.4	Tag Disposition Code to Call ID	254
5.6.5	Escalate to Supervisor.....	255
5.6.6	Emergency Call to Supervisor	258
5.7	Meet-Me Conference Requests (AS only)	261
5.7.1	Get Meet-Me Conference.....	261
5.7.2	Outgoing Dial	262
5.7.3	Mute Call	265
5.7.4	Unmute Call	266
5.7.5	Hold Call.....	267
5.7.6	Retrieve Call.....	268
5.7.7	Release Call.....	269
5.7.8	Lock	270
5.7.9	Unlock.....	271
5.7.10	Start Recording	272
5.7.11	Pause Recording	273
5.7.12	Resume Recording.....	274
5.7.13	Stop Recording	275
5.7.14	Set Lecture Mode	276
5.7.15	Unset Lecture Mode	277
5.7.16	Play Security Classification	278
5.7.17	Set Conference Client Session Info.....	279



5.8	Collaborate Room Requests (AS only)	280
5.8.1	Get Collaborate Room.....	280
5.8.2	Outgoing Dial	281
5.8.3	Mute Call	283
5.8.4	Unmute Call	284
5.8.5	Release Call.....	285
5.8.6	Lock	286
5.8.7	Unlock.....	287
5.8.8	Set Conference Client Session Info.....	288
5.9	Service Management Requests	289
5.9.1	Modify Call Forwarding Always.....	289
5.9.2	Get Call Forwarding Always.....	290
5.9.3	Modify Call Forwarding No Answer	291
5.9.4	Get Call Forwarding No Answer	292
5.9.5	Modify Call Forwarding Busy	293
5.9.6	Get Call Forwarding Busy	294
5.9.7	Modify Hoteling Guest	295
5.9.8	Get Hoteling Host List (AS only)	296
5.9.9	Get Hoteling Guest (AS only).....	297
5.9.10	Modify Do Not Disturb	298
5.9.11	Get Do Not Disturb	298
5.9.12	Clear Voice Mail Message Summary	299
5.9.13	Get Voice Mail Message Summary	300
5.10	Event Channel Requests	301
5.10.1	Add Channel	301
5.10.2	Update Channel.....	302
5.10.3	Delete Channel	303
5.10.4	Get Channel Set	304
5.11	Event Subscription Requests	305
5.11.1	Add User Subscription.....	305
5.11.2	Add SubUser Subscription	306
5.11.3	Add Group Subscription (AS only).....	308
5.11.4	Add Enterprise Subscription (AS only)	309
5.11.5	Add Service Provider Subscription (AS only).....	311
5.11.6	Add System Subscription (AS only).....	312
5.11.7	Update Subscription	313
5.11.8	Delete Subscription.....	314
5.11.9	Get Single Subscription	315
5.11.10	Get Subscription	316
5.12	Application Controller Request (AS only)	318
5.12.1	Set Application Controller State	318
5.12.2	Get Application Controller State	319

5.13	Voice Messaging (AS only)	320
5.13.1	Delete Voice Messaging Message Request	320
5.13.2	Get Voice Messaging Message Request	320
5.13.3	Get All Voice Messaging Messages Request	321
5.13.4	Mark All Voice Messaging Messages as Unread Request	322
5.13.5	Mark All Voice Messaging Messages as Read Request	323
5.13.6	Mark Voice Messaging Message as Unread Request	324
5.13.7	Mark Voice Messaging Messages As Read Request	325
6	Events	326
6.1	Event and Event Response	326
6.1.1	Event Format	326
6.1.2	Event Response	327
6.2	Call Events	328
6.2.1	Call Originating	329
6.2.2	Call Originated	329
6.2.3	Call Received	330
6.2.4	Call Answered	332
6.2.5	Call Held	333
6.2.6	Call Retrieved	335
6.2.7	Call Updated	336
6.2.8	Call Redirected	338
6.2.9	Call Transferred	340
6.2.10	Call Releasing	341
6.2.11	Call Released	343
6.2.12	Call Collecting	344
6.2.13	Call Picked-up	345
6.2.14	Call Barged-in	347
6.2.15	Call Monitored (AS only)	348
6.2.16	Call Forwarded	350
6.2.17	Call Detached	351
6.2.18	Call Park Retrieved	352
6.2.19	Call Recording Starting (AS only)	354
6.2.20	Call Recording Started (AS only)	356
6.2.21	Call Recording Paused (AS only)	357
6.2.22	Call Recording Resumed (AS only)	359
6.2.23	Call Recording Stopped (AS only)	361
6.2.24	Call Subscription and Call Subscription Resync (AS only)	363
6.2.25	Call Security Classification Updated (AS only)	365
6.2.26	Call Client Session Info Updated (AS only)	367
6.2.27	Hook Status (AS only)	368
6.3	Conference Events	369
6.3.1	Conference Started	369



6.3.2	Conference Held	370
6.3.3	Conference Retrieved.....	371
6.3.4	Conference Updated	372
6.3.5	Conference Released.....	372
6.3.6	Conference Muted	373
6.3.7	Conference Unmuted	373
6.3.8	Conference Call Muted.....	374
6.3.9	Conference Call Unmuted.....	374
6.3.10	Conference Call Made Deaf.....	375
6.3.11	Conference Call Made Undeaf	376
6.4	Route Point Events (AS only).....	376
6.4.1	Route Point Call Added	377
6.4.2	Route Point Call Offered to Agent.....	378
6.4.3	Route Point Call Answered by Agent	379
6.4.4	Route Point Call Abandoned.....	380
6.4.5	Route Point Call Overflowed	381
6.4.6	Route Point Call Overflowed Treatment Completed.....	382
6.4.7	Route Point Call Transferred.....	383
6.4.8	Route Point Call Updated.....	383
6.4.9	Route Point Call Bounced	384
6.4.10	Route Point Call Forwarded	385
6.4.11	Route Point Call Forwarded Treatment Completed.....	386
6.4.12	Route Point Call Released	386
6.4.13	Route Point Call Failed	387
6.4.14	Route Point Outgoing Call Originated	388
6.4.15	Route Point Outgoing Call Answered	389
6.4.16	Route Point Subscription Event and Route Point Subscription Resync Event	390
6.4.17	Route Point Holiday Policy Applied	391
6.4.18	Route Point Holiday Policy Treatment Completed.....	391
6.4.19	Route Point Night Policy Applied	392
6.4.20	Route Point Night Policy Treatment Completed	393
6.4.21	Route Point Treatment Started	393
6.4.22	Route Point Treatment Completed	394
6.4.23	Route Point MOH Started	395
6.4.24	Route Point MOH Completed	396
6.4.25	Route Point Ringback Started.....	397
6.4.26	Route Point Ringback Completed	398
6.4.27	Route Point Busy Started	399
6.4.28	Route Point Busy Completed	400
6.4.29	Route Point Silence Started	401
6.4.30	Route Point Silence Completed	401
6.4.31	Route Point Whisper Started.....	402



6.4.32 Route Point Failed	403
6.4.33 Route Point Recovered	403
6.5 ACD Events (AS only)	404
6.5.1 ACD Call Added.....	404
6.5.2 ACD Call Offered to Agent	406
6.5.3 ACD Call Answered by Agent.....	407
6.5.4 ACD Whisper Started	408
6.5.5 ACD Call Overflowed.....	409
6.5.6 ACD Call Overflowed Treatment Completed	410
6.5.7 ACD Call Abandoned	412
6.5.8 ACD Call Transferred	413
6.5.9 ACD Call Updated	414
6.5.10 ACD Call Bounced.....	415
6.5.11 ACD Call Forwarded.....	416
6.5.12 ACD Call Forwarded Treatment Completed	417
6.5.13 ACD Call Released.....	417
6.5.14 ACD Call Reordered.....	418
6.5.15 ACD Call Promoted	419
6.5.16 ACD Call Stranded	420
6.5.17 ACD Call Stranded Treatment Completed	421
6.5.18 ACD Call Stranded Unavailable	423
6.5.19 ACD Call Stranded Unavailable Treatment Completed	424
6.5.20 ACD Call Escaped.....	425
6.5.21 ACD Subscription Event and ACD Subscription Resync Event	426
6.5.22 ACD Holiday Policy Applied	427
6.5.23 ACD Holiday Policy Treatment Completed	428
6.5.24 ACD Night Policy Applied	429
6.5.25 ACD Night Policy Treatment Completed	429
6.6 Agent Events (AS only)	430
6.6.1 Agent State.....	430
6.6.2 Agent Disposition Code Added.....	431
6.6.3 ACD Agent Join Update	432
6.6.4 Agent Subscription.....	432
6.7 Agent Monitoring Events (AS only).....	432
6.7.1 Call Center Agent Monitoring	432
6.7.2 Call Center Agent On Call Alert	433
6.7.3 Call Center Agent Monitoring Subscription	433
6.8 ACD Configuration Events (AS only).....	434
6.8.1 ACD Forced Forwarding	434
6.8.2 ACD Holiday Service	434
6.8.3 ACD Night Service	435
6.8.4 ACD Configuration Subscription	435



6.9	ACD Monitoring Events (AS only).....	435
6.9.1	Call Center Monitoring	435
6.10	Call Park Events	436
6.10.1	Call Parked Against	436
6.11	Meet-Me Conference Events (AS only).....	437
6.11.1	Meet-Me Conference Started.....	438
6.11.2	Meet-Me Conference Stopped.....	438
6.11.3	Meet-Me Conference Locked	439
6.11.4	Meet-Me Conference Unlocked.....	440
6.11.5	Meet-Me Conference Recording Started	440
6.11.6	Meet-Me Recording Stopped.....	441
6.11.7	Meet-Me Recording Paused	442
6.11.8	Meet-Me Recording Resumed.....	442
6.11.9	Meet-Me Lecture Mode Started.....	443
6.11.10	Meet-Me Lecture Mode Stopped	444
6.11.11	Meet-Me Auto Lecture Mode Started	444
6.11.12	Meet-Me Participant Joined.....	445
6.11.13	Meet-Me Participant Left	446
6.11.14	Meet-Me Participant Muted	446
6.11.15	Meet-Me Participant Unmuted	447
6.11.16	Meet-Me Participant Held	448
6.11.17	Meet-Me Participant Retrieved.....	448
6.11.18	Meet-Me Participant Updated	449
6.11.19	Meet-Me Conference Subscription Event and Meet-Me Conference Resync Subscription Event	450
6.11.20	Meet-Me Conference Security Classification Updated	450
6.11.21	Meet-Me Conference Client Session Info Updated	451
6.12	Collaborate Room Events (AS only).....	451
6.12.1	Collaborate Room Started	452
6.12.2	Collaborate Room Stopped.....	452
6.12.3	Collaborate Room Locked.....	453
6.12.4	Collaborate Room Unlocked	453
6.12.5	Collaborate Room Participant Joined	453
6.12.6	Collaborate Room Participant Left	454
6.12.7	Collaborate Room Participant Muted.....	454
6.12.8	Collaborate Room Participant Unmuted	455
6.12.9	Collaborate Room Participant Updated	455
6.12.10	Collaborate Room Subscription Event and Collaborate Room Resync Subscription Event	456
6.12.11	Collaborate Room Client Session Info Updated	457
6.13	Service Management Events	457
6.13.1	Call Forwarding Always.....	457
6.13.2	Call Forwarding No Answer	458

6.13.3 Call Forwarding Busy	458
6.13.4 Do Not Disturb.....	459
6.13.5 Call Transfer.....	459
6.13.6 Call Waiting	460
6.13.7 Last Number Redial.....	460
6.13.8 Music On Hold	461
6.13.9 N-Way Call	461
6.13.10 Third-Party Voice Mail Support	461
6.13.11 Three-Way Call	462
6.13.12 Voice Messaging.....	462
6.13.13 Sequential Ring.....	462
6.13.14 Simultaneous Ring Personal.....	463
6.13.15 Voice Mail Message Summary	464
6.13.16 Hoteling Guest (AS only).....	464
6.13.17 BroadWorks Anywhere.....	464
6.13.18 Remote Office	465
6.13.19 CommPilot Express	465
6.13.20 Flexible Seating.....	466
6.13.21 Personal Assistant	467
6.14 Subscription Event.....	467
6.14.1 Subscription Terminated	467
6.15 Channel Event	467
6.15.1 Channel Terminated	467
7 XML Schema.....	469
7.1 XspXMLInterface Schema	469
7.2 CTI Schema	469
7.2.1 URI Encoding	470
7.3 Xtended Services Interface Schema	471
7.4 XML Examples.....	472
7.4.1 Request	472
7.4.2 Response	472
7.4.3 Event.....	473
7.4.4 EventResponse	473
7.4.5 Keepalive.....	474
7.4.6 KeepaliveResponse.....	474
7.5 Character Sets Encoding	474
7.5.1 Incoming Message	474
7.5.2 Outgoing Message	474
8 Type Definitions	475
8.1 Message Parameter Type Definitions	475
8.1.1 Basic Types.....	475
8.1.2 Advanced Types	475



8.1.3	Common Types	476
8.2	Event Channel	477
8.2.1	Channel	477
8.2.2	ChannelResponse	478
8.2.3	ChannelTerminatedReason	478
8.3	Subscription	479
8.3.1	Subscription.....	479
8.3.2	SubscriptionId	480
8.3.3	SubscriptionList.....	480
8.3.4	EntityType	480
8.3.5	EventPackage.....	480
8.4	Common Request and Response Related Types	482
8.4.1	SessionId.....	482
8.4.2	RequestId	482
8.4.3	EventId	482
8.4.4	EventResponseStatus.....	482
8.4.5	ExternalApplicationId	482
8.5	Common Types	482
8.5.1	Address	482
8.5.2	AccessEndpoint	483
8.5.3	PartyInformation.....	483
8.5.4	UserId	485
8.6	Call Related Types	485
8.6.1	HookStatus.....	485
8.6.2	Call.....	485
8.6.3	CallInfoList.....	496
8.6.4	CallInfo.....	496
8.6.5	CallIdList.....	496
8.6.6	CallList	496
8.6.7	CallRecordingMode	496
8.6.8	CallStartInfo.....	497
8.6.9	CallMeNowStartInfo.....	497
8.6.10	CallMeNowPasscode	497
8.6.11	ClickToDialLocationEnum	497
8.6.12	IMRNInfo	497
8.7	Conference Related Types	498
8.7.1	Conference.....	498
8.8	Route Point Related Types	499
8.8.1	RoutePointQueue	499
8.8.2	RoutePointQueueEntryList.....	499
8.8.3	RoutePointQueueEntry	499
8.8.4	RoutePointState.....	500



8.8.5	StateChangeReason	501
8.8.6	OverflowReason	501
8.8.7	RoutePointCallFailureReason.....	501
8.8.8	RoutePointCallReleaseReason	501
8.8.9	CPDResult	502
8.8.10	NonEmptyURI	502
8.8.11	RoutePointCallStartInfo	502
8.8.12	Media Playback and Digit Collection Types	502
8.8.13	RoutePointDistribute.....	505
8.8.14	RoutePointPlayTreatment.....	505
8.8.15	RoutePointPlayMusicOnHold	506
8.8.16	RoutePointPlayRingback	506
8.8.17	RoutePointPlaySilence	506
8.8.18	RoutePointPlayBusy.....	507
8.8.19	RoutePointOutgoingDial.....	507
8.9	ACD	507
8.9.1	ACDQueue.....	507
8.9.2	ACDQueueEntryList	507
8.9.3	ACDQueueEntry	507
8.9.4	ACDQueueEntryList	509
8.9.5	QueueCallOrder.....	509
8.9.6	ACDPromote.....	509
8.9.7	CallCenterMonitoringStatus	510
8.10	Agent	511
8.10.1	AgentStateInformation.....	511
8.10.2	AgentUnavailableCode.....	513
8.10.3	AgentDispositionCode	513
8.10.4	ACDAgentJoinUpdateData	513
8.10.5	ACDAgentJoinInfo	514
8.10.6	ACDAgentJoinData	514
8.10.7	ACDAgentJoinInfoList	514
8.10.8	Call Center	514
8.10.9	CallCenterAgentMonitoringStatus	515
8.10.10	CallCenterAgentOnCallAlertInfo	515
8.11	Meet-Me Conference Related Types	516
8.11.1	MeetMeConference	516
8.12	Collaborate Room Related Types	518
8.12.1	CollaborateRoom.....	518
8.13	Service Management	519
8.13.1	CallForwardingAlways	519
8.13.2	CallForwardingNoAnswer	519
8.13.3	CallForwardingBusy	520

8.13.4	DoNotDisturb.....	520
8.13.5	BroadWorksAnywhereInfo	520
8.13.6	ActivatableServiceInfo	521
8.13.7	CallTransferInfo.....	522
8.13.8	FlexibleSeatingGuest	522
8.13.9	SequentialRing.....	524
8.13.10	SimultaneousRingPersonal.....	525
8.13.11	VoiceMailMessageSummary	526
8.13.12	Remote Office	526
8.13.13	CommPilotExpress	526
8.13.14	HotelingGuest	530
8.13.15	AvailableHotelingHosts.....	530
8.13.16	UserIdList	530
8.13.17	PersonalAssistant	531
8.14	Application Controller.....	532
8.14.1	ApplicationControllerState	532
8.15	Voice Messaging	532
8.15.1	VoiceMessagingMessages	532
8.15.2	VoiceMessage	534
9	Message Flows.....	535
9.1	Call Message Flows	535
9.1.1	Dial.....	536
9.1.2	Hold	545
9.1.3	Retrieve	548
9.1.4	Release	552
9.1.5	Consultative Call	556
9.1.6	Call Forwarding	567
9.1.7	Releasing	574
9.1.8	Blind Transfer.....	577
9.1.9	Consultative Transfer.....	583
9.1.10	Mute Transfer.....	590
9.1.11	Reconnect	598
9.1.12	Transfer to Voice Mail.....	604
9.1.13	Call Return (Two-level Mode)	609
9.1.14	Directed Call Pickup	620
9.1.15	Directed Call Pickup with Barge-in.....	630
9.1.16	Monitor Call (AS only).....	644
9.2	Conference.....	658
9.2.1	Conference Start.....	658
9.2.2	Conference Call Muted and Made Deaf.....	665
9.2.3	Conference Released.....	672
9.2.4	Conference Held.....	682



9.2.5	Conference Retrieved.....	684
9.2.6	Conference Add Participant	686
9.3	Route Point Message Flows (AS only).....	691
9.3.1	Call Received with Treatment and Then Distributed	692
9.3.2	Transfer	707
9.3.3	Bounced and Abandoned Call	710
9.3.4	Release Call.....	721
9.3.5	Call Updated	724
9.3.6	Fail Call.....	726
9.3.7	Call Overflow.....	729
9.3.8	Call Forwarded.....	732
9.3.9	Holiday Policy Applied	734
9.3.10	Monitor Next Route Point	737
9.3.11	Outgoing Call Distributed to Agent.....	754
9.3.12	Route Point Failure and Recovery.....	766
9.4	ACD Message Flows (AS only)	773
9.4.1	Call Received with Treatment then Distributed.....	774
9.4.2	Bounced and Abandoned Call	784
9.4.3	Call Overflow	792
9.4.4	Call Forwarded.....	795
9.4.5	Holiday Policy Applied	797
9.4.6	Call Stranded	800
9.4.7	Call Transfer.....	802
9.4.8	Call Updated	806
9.4.9	Call Promoted, Reordered and Finally Escaped.....	807
9.5	Agent Message Flows (AS only).....	812
9.5.1	Agent Join and State Change	813
9.5.2	Escalate to Supervisor.....	815
9.5.3	Emergency Call to Supervisor	824
9.6	Service Management Message Flows	832
9.6.1	Call Forwarding Always.....	833
9.6.2	Do Not Disturb.....	834
9.6.3	Hoteling Host and Guest (AS only).....	836
9.6.4	Voice Mail Message Summary (AS only).....	839
10	Error Code List.....	841
11	Acronyms and Abbreviations.....	849
12	References	851

Table of Figures

Figure 1 High-level View.....	31
Figure 2 Message Exchange	34
Figure 3 High-level View of Requests	35
Figure 4 High-level View of Events.....	37
Figure 5 Active-Standby with Synchronization.....	38
Figure 6 Active-Active with Synchronization	39
Figure 7 Active-Active Without Synchronization	40
Figure 8 Initialization	52
Figure 9 Call Session.....	57
Figure 10 Subscriber to Network Party Call Session.....	58
Figure 11 Access Endpoint	67
Figure 12 Network Party.....	68
Figure 13 Intragroup/Enterprise Topology.....	69
Figure 14 Intra-Cisco BroadWorks Topology.....	69
Figure 15 Redirections	69
Figure 16 Extra-Cisco BroadWorks Topology.....	70
Figure 17 Single Call Topology	70
Figure 18 Call Session Redirection	71
Figure 19 Call Session Forked.....	71
Figure 20 Executive Call Topology	72
Figure 21 Conference.....	72
Figure 22 Incoming Call Processing.....	75
Figure 23 Media Playback and Digits Collection.....	83
Figure 24 Meet-Me Conference	91
Figure 25 Subscriber S1 Connected to the Bridge but not Entered the Conference PIN	92
Figure 26 Collaborate Room.....	96
Figure 27 Subscriber S1 Dialed into the Collaborate Room but not Entered the Room ID.....	97
Figure 28 Simple Dial	111
Figure 29 Dial with Existing Active Call	112
Figure 30 Call Answered Through Talk Request.....	115
Figure 31 Call Answered Through Talk Request.....	116
Figure 32 Talk on Incoming Alerting Call.....	117
Figure 33 Talk on Held Call (Held via Hold Request).....	118
Figure 34 Talk on Held Call (Held via Endpoint).....	118
Figure 35 Talk on Click-To-Dial Incoming Call.....	119
Figure 36 Talk on Held Call with One Active Call	119
Figure 37 Talk on Held Call with One Alerting Call (Personality Terminator).....	120
Figure 38 Talk on Held Call with One Alerting Call (Personality Originator)	120
Figure 39 Talk on Held Call with One Alerting Call (Personality Originator) Established Through Dial Request.....	121
Figure 40 Talk on Executive's Call.....	121
Figure 41 Hold Active Call.....	123
Figure 42 Hold Remote Held Call	124
Figure 43 Executive's Hold.....	124
Figure 44 Release Call	126
Figure 45 Release Call without Decline Option	126
Figure 46 Release Call Participating in Conference (Endpoint-Initiated Cisco BroadWorks Conference)	127
Figure 47 Release Call Participating in Conference (Cisco BroadWorks Conference).....	127
Figure 48 Release Call Participating in Conference (Three Participants)	128
Figure 49 Executive's Release	128
Figure 50 Blind Transfer State Changes	130



Figure 51 Executive Blind Transfer State Changes.....	131
Figure 52 Consultative Transfer State Changes.....	134
Figure 53 Mute Transfer State Changes	138
Figure 54 Reconnect State Changes	141
Figure 55 Transfer to Voice Mail	143
Figure 56 Call Return.....	146
Figure 57 Call Return Events	148
Figure 58 Directed Call Pickup.....	151
Figure 59 Directed Call Pickup Events	153
Figure 60 Directed Call Pickup with Barge-in.....	156
Figure 61 Directed Call Pickup with Barge-in State Changes	157
Figure 62 Events	159
Figure 63 Monitor Call	162
Figure 64 Monitor Call State Changes	163
Figure 65 Monitor Call State Changes	165
Figure 66 Monitor Next Call.....	169
Figure 67 Monitor Next Call State Changes.....	170
Figure 68 Releasing Monitor Next Call	171
Figure 69 Releasing Monitor Next Call State Changes.....	172
Figure 70 Monitor Next Call on Agent	174
Figure 71 Monitor Next Call onAgent Events	175
Figure 72 Call Me Now	183
Figure 73 Retrieve	187
Figure 74 Find-me/Follow-me Call Push	190
Figure 75 Find-me/Follow-me Call Push Events	191
Figure 76 Executive Assistant Initiate Call	194
Figure 77 Events	195
Figure 78 Executive-Assistant Call Push	196
Figure 79 Executive Assistant Call Push Associated Events.....	197
Figure 80 Conference Start.....	204
Figure 81 Conference Release State Changes.....	206
Figure 82 Conference Hold State Changes	207
Figure 83 Conference Retrieve State Changes	209
Figure 84 Conference Mute State Changes	210
Figure 85 Conference Unmute State Changes	212
Figure 86 Conference Add Participant State Changes	213
Figure 87 Distribute Call State Changes	220
Figure 88 Route Point Distribute Call Events	221
Figure 89 Route Point Distribute Call	222
Figure 90 Outgoing Dial State Changes	231
Figure 91 Bounced Call is Put Back in Route Point Queue State Changes	234
Figure 92 Bounced Call is Transferred to New Destination State Changes	235
Figure 93 Blind Transfer State Changes	237
Figure 94 Release Queued Call	239
Figure 95 Release Offered Call	239
Figure 96 Fail Call State Changes	241
Figure 97 Call in Queue.....	243
Figure 98 Call in Queue but Offered to Agent.....	244
Figure 99 Blind Transfer State Changes	246
Figure 100 Release Queued Call	250
Figure 101 Release Offered Call	251
Figure 102 Escalate to Supervisor State Changes	256
Figure 103 Emergency Call to Supervisor State Changes.....	259
Figure 104 Emergency Call to Supervisor Events	260
Figure 105 Meet-Me Outgoing Dial	263

Figure 106 Meet-Me Conference Outgoing Dial Events for a Started Conference	264
Figure 107 Meet-Me Conference Outgoing Dial Events for a New Conference.....	264
Figure 108 Meet-Me Release Call.....	269
Figure 109 Collaborate Outgoing Dial	281
Figure 110 Collaborate Room Outgoing Dial Events for a Started Conference	282
Figure 111 Collaborate Room Outgoing Dial Events for a New Conference.....	282
Figure 112 Collaborate Room Release Call	285
Figure 113 Dial Message Flow	536
Figure 114 Hold Message Flow.....	545
Figure 115 Retrieve Message Flow.....	548
Figure 116 Release Message Flow.....	552
Figure 117 Consultative Call Message Flow.....	556
Figure 118 Call Forwarding Message Flow.....	567
Figure 119 Releasing Message Flow	574
Figure 120 Blind Transfer Message Flow	577
Figure 121 Consultative Transfer Message Flow	583
Figure 122 Mute Transfer Message Flow	590
Figure 123 Reconnect Message Flow	598
Figure 124 Transfer to Voice Mail Message Flow	604
Figure 125 Call Return Message Flow	609
Figure 126 Directed Call Pickup Message Flow	620
Figure 127 Directed Call Pickup with Barge-in Message Flow	630
Figure 128 Monitor Call Message Flow.....	644
Figure 129 Conference Start Message Flow	658
Figure 130 Conference Call Muted and Made Deaf Message Flow	665
Figure 131 Conference Released Message Flow	672
Figure 132 Conference Held Message Flow.....	682
Figure 133 Conference Retrieved Message Flow	684
Figure 134 Conference Add Participant Message Flow.....	686
Figure 135 Call Received with Treatment and Then Distributed	692
Figure 136 Transfer Message Flow.....	707
Figure 137 Bounced and Abandoned Call Message Flow.....	710
Figure 138 Release Call Message Flow	721
Figure 139 Call Updated Message Flow	724
Figure 140 Fail Call Message Flow	726
Figure 141 Call Overflow Message Flow	729
Figure 142 Call Forwarded Message Flow	732
Figure 143 Holiday Policy Applied Message Flow.....	734
Figure 144 Monitor Next Route Point Message Flow	737
Figure 145 Outgoing Call Distributed to Agent.....	754
Figure 146 Route Point Failure and Recovery Message Flow	766
Figure 147 Call Received with Treatment and Then Distributed	774
Figure 148 Bounced and Abandoned Call Message Flow.....	784
Figure 149 Call Overflow Message Flow	792
Figure 150 Call Forwarded Message Flow	795
Figure 151 Holiday Policy Applied Message Flow.....	797
Figure 152 Call Stranded Message Flow	800
Figure 153 Call Transfer Message Flow	802
Figure 154 Call Updated Message Flow	806
Figure 155 Call Promoted, Reordered, and Finally Escaped Message Flow	807
Figure 156 Call Promoted, Reordered, and Finally Escaped Message Flow	815
Figure 157 Emergency Call to Supervisor Message Flow	824
Figure 158 Clear Voice Mail Message Summary	839

1 Summary of Changes

This section describes the changes to this document for each release and document version.

1.1 Changes for Release 23.0, Document Version 4

This version of the document includes the following change:

- Added section [7.2.1 URI Encoding](#) for PR-58772.

1.2 Changes for Release 23.0, Document Version 3

This version of the document includes the following change:

- Rebranded product name for Cisco.

1.3 Changes for Release 23.0, Document Version 2

This version of the document includes the following changes:

- Introduced Release 23.0-specific functionality:
 - Simplify Notifications to UCaaS and Collaborate (FR 13709)
 - Call Through Support Of Collaborate Bridge ID (FR 11764)

1.4 Changes for Release 22.0, Document Version 1

This version of the document includes the following changes:

- Introduced Release 22.0-specific functionality:
 - Mute For Conference Controller (FR 7186)
 - Call Through for BroadWorks Anywhere (FR 7406)

1.5 Changes for Release 21.0, Document Version 3

This version of the document includes the following changes:

- Removed maxChannelsReached reason from channelTerminatedEvent in section [8.2.3 ChannelTerminatedReason](#) for PR-47018.
- Clarified system domain usage for the Get Agent State request in section [5.6.1.1 Description](#) for PR-45563.

1.6 Changes for Release 21.0, Document Version 2

This version of the document includes the following change:

- Updated server icons.

1.7 Changes for Release 21.0, Document Version 1

This version of the document includes the following changes:

- Introduced Release 21.0-specific functionality:
 - Personal Assistant (FR 204023)
 - Mobility Personal Management (FR 210489)
 - MP3 Support For Voice Messaging Service (FR 190717)



- Collaborate (FR 214095)

1.8 Changes for Release 20.0, Document Version 2

This version of the document includes the following change:

- Added section [5.4.3 Play Treatment](#) that was missing from the document for EV 218655.

1.9 Changes for Release 20.0, Document Version 1

This version of the document includes the following changes:

- Introduced Release 20.0 specific functionality:
 - Executive and Assistant Services (FR 160554)
 - Xtended Services Interface (Xsi) Subscription Refresh on Application Server Failure (FR 167781)
 - Access Link Counting Enhancements (FR 166936)
 - Call Recording – Start/Stop/Pause/Resume User Control (FR 170972)
 - Extended Away (FR 173740)
 - Voice Mail Interface on Xtended Services Platform for BroadCloud (FR 155575)
 - Visual Security Classification for Active Call (FR 170974)
 - Business Communicator-Chat and Media Session Correlation (FR 178319)
 - Flexible Seating Service (FR 114601)
 - Mobility Service Enhancement (FR 181182)
 - Hook Status for All Call Xsi Event Packages (FR 193123)
- Updated section [3.6 Backward Compatibility](#) for EV 185622.
- Updated section [6.10 Call Park Events](#) for EV 193880.
- Updated section [5.7.2 Outgoing Dial](#) for EV 197892.
- Added section [7.5 Character Sets Encoding](#) for EV 159081.
- Updated section [6.11 Meet-Me Conference Events \(AS only\)](#) for EV 200958.
- Update section for [6.2 Call Events](#) for EV 187391.

1.10 Changes for Release 19.0, Document Version 1

This version of the document includes the following changes:

- Introduced Release 19.0 specific functionality:
 - Call Center thresholds and alerts (FR 150062)
 - Find-Me/Follow-Me (FR 125258)
 - Call-Center Skill-Based routing (FR 149947)
 - Call-Center Stranded Unavailable policy (FR 149564)
 - Shared Subscription mechanism (FR 160418)
 - Change to channel create and termination (FR 163735)
- Updated keepalive description in section [3.4 Events](#).

- Added Meet-Me conferencing functionality.
- Updated section [5.9.7 Modify Hoteling Guest](#) for EV 167204.
- Updated section [5.6 Agent Requests \(AS only\)](#) for EV 174309.
- Updated document for EVs 159752, 159750, 163827, and 131756.

1.11 Changes for Release 18.0, Document Version 2

This version of the document includes the following changes:

- Updated sections [6.1 Event and Event Response](#) and [6.15.1 Channel Terminated](#) for EV 146293.

1.12 Changes for Release 18.0, Document Version 1

There were no changes specific to Release 18.0.

1.13 Changes for Release 17.sp4, Document Version 1.12

The following Call requests were added:

- Record
- Retrieve

The following Call requests were modified:

- Dial
- Call Return
- Directed Call Pickup
- Directed Call Pickup with Barge-In
- Last Number Redial
- Monitor Call
- Monitor Next Call

The following Agent request was modified:

- Escalate Call to Supervisor

The following event packages were added:

- Call Recording

The following event packages were modified:

- Basic Call
- Standard Call
- Advanced Call

1.14 Changes for Release 17.sp3, Document Version 1.12

The following Call requests were added:

- Conference Unmute request.
- Call Me Now request.

The following event packages were added:

- Call Park

The following event packages were modified:

- Call Center Agent
- Call Center (ACD) Configuration
- Call Center (ACD) Queue
- Route Point Queue

1.15 Changes for Releases 17.sp2, Document Version 1.12

The following event packages were added:

- Call Center Agent Monitoring
- Call Center Monitoring

1.16 Changes for Release 17.0, Document Version 1.11

This version of the document includes the following change:

- Updated section [5.12 Application Controller Request](#) for EV 117612.

1.17 Changes for Release 17.0, Document Version 1.10

This version of the document includes the following changes:

- Updated section [6.3.1 Conference Started](#) for EV 130182.
- Added section [10 Error Code List](#) for EV 120600.

1.18 Changes for Release 17.0, Document Version 1.9

This version of the document includes the following change:

- Updated section [5 Requests](#) for EV 121906.

1.19 Changes for Release 17.0, Document Version 1.8

This version of the document includes the following change:

- Updated section for EV 111969.

1.20 Changes for Release 17.0, Document Version 1.7

This version of the document includes the following changes:

- Updated sections [3 System Concepts](#), [4 Call and Service Management Concepts](#), and [5 Requests](#) for EV 120600.

1.21 Changes for Release 17.0, Document Version 1.6

This version of the document includes the following changes:

- Updated section [5 Requests](#) for EVs 117895 and 117612.

1.22 Changes for Release 17.0, Document Version 1.5

This version of the document includes the following changes:

- Updated section [6.2.2 Call Originated](#) with new error codes for EV 113350.

- Added error codes.
- Repaired hyperlinks in the document.

1.23 Changes for Release 17.0, Document Version 1.4

This version of the document includes the following changes:

- Updated section [6.2.2 Call Originated](#) for EV 111837, section [6.2.5 Call Held](#) for EV 111900, and section [6.2.6 Call Retrieved](#) for EV 111902.
- Updated event package name, *Monitor Next Call* request description and various message flows based on latest implementation.
- Updated section [5 Requests](#) for EV 112516, section [6.5 ACD Events](#) for EV 112367. Clarified *Route Point* event completion in section [4.3.6 Media Playback and Digits Collection](#).
- Updated section [3.4.5.5 Target Scope Rules](#) for EV 112654.

1.24 Changes for Release 17.0, Document Version 1.3

This version of the document includes the following changes:

- Made minor editorial changes.

1.25 Changes for Release 17.0, Document Version 1.2

This version of the document includes the following changes:

- Added service event package in section [3.4.5.8 Event Package](#) and described associated event description in section [6.13 Service Management Events](#).
- Introduced new section [7 XML Schema](#) that provides a description of the XML schema.
- Updated section using captured traces.

1.26 Changes for Release 17.0, Document Version 1.1

This version of the document includes the following changes:

- Updated section [3.6 Backward Compatibility](#) to provide additional information.
- Provided a list of requests that modifies the external tracking ID and clarified the impact.
- Fixed various discrepancies.
- Clarified how the device capability can impact call establishment in section [5.2.1.6 Associated Events](#).
- Updated section [5.4.8 Outgoing Dial](#).
- Updated section [6.4.8 Route Point Call Updated](#).
- Updated section [6.4 Route Point Events \(AS only\)](#) and [6.4.16 Route Point Subscription Resync Event](#).
- Added various error cases and fixed minor errors.
- Aligned types and parameter names to reflect the implementation.
- Updated section [5.6.6 Emergency Call to Supervisor](#) to reflect the implementation.



2 Overview

2.1 Purpose

This document is the specification for a Computer Telephony Integration (CTI) interface into Cisco BroadWorks. The Computer Telephony Integration interface (hereafter referred to as the CTI interface) allows remote applications to integrate with Cisco BroadWorks to perform telephony-related actions and to be notified about telephony events against subscribers hosted on a Cisco BroadWorks-powered network.

The Cisco BroadWorks CTI interface allows applications to control a subscriber's telephony experience by supporting various kinds of call control requests and notifications. The interface defines a rich set of requests for call control as well as services, and publishes/generates events that are used to inform applications about both call and service related activity that occurs on subscribers.

For the remainder of this document, the "CTI interface" and "interface" are used interchangeably, as are the terms "remote application" and "external application".

2.1.1 Document Structure

Each section in this specification document serves a distinct purpose:

- Section [3 System Concepts](#) presents system/high-level concepts and message interaction examples essential in understanding how an application would integrate with Cisco BroadWorks using the CTI interface. Nomenclature such as Requests/Responses, Events and Event Delivery, High Availability, and Backward Compatibility are introduced in this section.
- Section [4 Call and Service Management Concepts](#) contains detailed explanations for the various interface element terms used throughout this document. Reading this section allows integrators to understand the glossary of requests that are sent to Cisco BroadWorks, responses returned by Cisco BroadWorks and events that are generated by Cisco BroadWorks.
- Section [5 Requests](#) describes the data elements in each request and associated responses supported by the Cisco BroadWorks CTI interface.
- Section [6 Events](#) describes the data elements in all events that are produced by Cisco BroadWorks over the CTI interface.
- Section [7 XML Schema](#) describes the structure of the XML schema.
- Section [8 Type Definitions](#) specifies CTI message element and parameter data types.
- Section [9 Message Flows](#) has message flows that document expected interactions between external applications and Cisco BroadWorks over the CTI interface. These flows lay out the data exchange required to perform supported actions, request for event generation, and process such event notifications.

2.2 Computer Telephony Integration Interface

2.2.1 Overview

The following diagram lays out a high-level view of one or more remote applications using the CTI interface into Cisco BroadWorks.

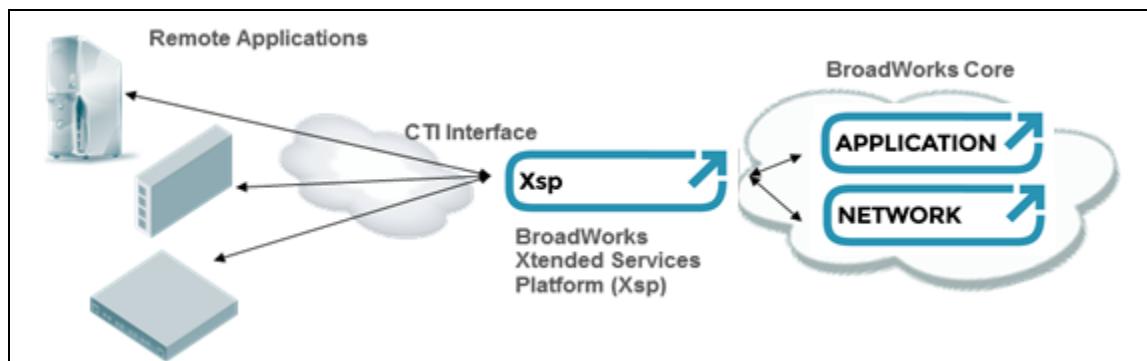


Figure 1 High-level View

The Cisco BroadWorks CTI interface is served by the Cisco BroadWorks Xtended Services Platform (Xsp) server. The Xtended Services Platform sits between CTI applications and the Cisco BroadWorks core network. Within the core Cisco BroadWorks network, the servers that the CTI interface uses to deliver its functions are the Application Server and Network Server.

A remote application creates one or more connections over the CTI interface. These connections are TCP or TLS socket connections opened against a Cisco BroadWorks Xtended Services Platform server. Over these connections, an application can send in requests and receive responses. Event delivery mechanisms are also established over these connections.

2.2.2 Cisco BroadWorks Application

Using the CTI interface assumes familiarity with various computer telephony terms and concepts. Within Cisco BroadWorks, these terms and concepts are described in detail in section

[3 System Concepts](#). At a high level, these concepts are:

- **Calls** – Calls within Cisco BroadWorks follow a certain model with call states, personalities, topologies, actions, events and so on. Understanding these, as well as call associations with phone numbers, subscribers, devices, and so on, are vital to using the CTI interface.
- **Route Points and Automatic Call Distribution** – Cisco BroadWorks has been structured to model a Route Point and Automatic Call Distribution (ACD), its associated policies and events, and so on.
- **Services** – Subscribers within Cisco BroadWorks can be given service capabilities such as Do Not Disturb, Call Forwarding Always, Hoteling, and Voice Messaging.

2.2.3 Functionality

CTI functions are essentially divided into two categories: Requests and Responses and Events. Requests and Responses consist of CTI commands issued by applications to Cisco BroadWorks. Events consist of discrete, asynchronous notifications issued out by Cisco BroadWorks informing applications of activity occurring on subscribers.



2.2.3.1 Requests and Responses

Every CTI request is a command initiated by an external application. Every request results in a CTI response by Cisco BroadWorks. Each command is a request for an action to be taken on a Cisco BroadWorks subscriber. These actions expose a broad spectrum of functionality to support a variety of applications. Types of request include:

- **Call Management** – Exposes real-time call control primitives, such as Dial, Answer, Hold, Transfer, and so on.
- **Call Status** – Exposes the real-time abilities to retrieve the list of active calls and determine the call state of those calls.
- **Service Management** – Exposes the ability to retrieve and configure services managed by the Cisco BroadWorks Application Server.

For more information on Requests and Responses supported over the CTI interface, see section [4 Call and Service Management Concepts](#).

2.2.3.2 Events

The CTI interface can inform external applications when activity occurs on Cisco BroadWorks-based subscribers. These notifications occur only when an application first subscribes for one or more events from a set of available event packages. Types of events include:

- **Call Events** – Provides notification of real-time call information on active calls.
- **Service Events** – Provides notification when service configuration has changed.
- **Route Point and ACD Events** – Provides notification of real-time information on incoming and queued calls.

For a complete description of each event available over the CTI interface, see section [6 Events](#).

2.2.4 CTI Messaging

The Cisco BroadWorks CTI interface is part of the Cisco Xtended Services Interface (Xsi).

The Xtended Services Interface is a family of interfaces that provides access into Cisco BroadWorks for multiple integration efforts and is designed to be used over multiple protocols by different types of applications. The CTI interface is part of this family of interfaces with an emphasis for real-time, high-availability, server-to-server communication for subscriber telephony control.

CTI messages are XML-based, with strict and enforceable element type descriptions and structure. The CTI interface also defines message exchange flows and sequences.

2.2.5 Cisco BroadWorks Deployment Mode

Depending on the deployment configuration, Cisco BroadWorks subscribers can be hosted on two types of Telephony Application Servers (TASs): an Execution Server (XS) or an Application Server (AS). The two server types can co-exist within the same network however not all services are supported by the Execution Server. For more information, see the *Cisco BroadWorks XS Mode Configuration Guide* [13].

This document describes the CTI interface functionality available for both the Application Server and Execution Server. As a result, some requests and events presented in this document might not be available depending on the server hosting the subscriber. For example, it is not possible to query the hoteling service configuration for user hosted on an Execution Server since this functionality is not available on an Execution Server. However, it is possible to query it on an Application Server (providing that the service has been assigned to the user).



Unless otherwise specified, the functionality described in this document is common to the Application Server and Execution Server. Application Server-specific functionality is identified with an **(AS only)** tag at the beginning of the section. There is currently no Execution Server-specific functionality related to the CTI interface. For a detailed description of the services supported by an Execution Server, see the *Cisco BroadWorks Feature Overview* [12].

3 System Concepts

3.1 Overview

The remote application connects to the CTI interface on Cisco BroadWorks through a farm of Xtended Services Platform (Xsp) servers. The Xtended Services Platform provides the entry point to the Cisco BroadWorks call and service control services defined in this specification.

The remote application opens a TCP socket connection to the Xtended Services Platform for the purpose of invoking requests and/or receiving events. A remote application starting up initially synchronizes its state with Cisco BroadWorks by setting up event channels and event subscriptions. Once the remote application has reached a synchronized state, then events sent by Cisco BroadWorks allow the remote application to maintain synchronization. The remote application sends requests to Cisco BroadWorks to perform call and service control.

The communication link between the remote application and Cisco BroadWorks is bi-directional, the remote application, and the Xtended Services Platform exchange XML payload in a request/response manner. The following diagram highlights the types of messages that are exchanged over the communication link between the remote application and the Xtended Services Platform.

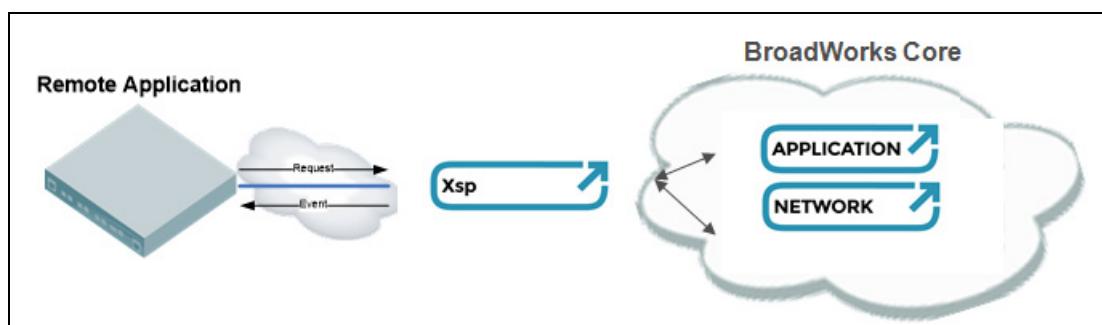


Figure 2 Message Exchange

The Xtended Services Platform is responsible for authenticating the requests received from the remote application. The Xtended Services Platform maintains a session to reduce the authentication overhead associated with processing a request from the remote application. The Xtended Services Platform is also responsible for providing request and response mapping and for routing events received from the Cisco BroadWorks core.

When a request is received from the remote application and successfully authenticated, then the request is processed by the Cisco BroadWorks core and an appropriate response is generated. Conversely, events are generated within the Cisco BroadWorks core and sent to the remote application through the Xtended Services Platform.

3.2 High-Availability Solution

The CTI interface defined in this specification enables the deployment of high-availability solutions. This specification allows both ends of the CTI interface to monitor the communication link and/or the state of synchronization of the remote application.

The Xtended Services Platform sends a handshake message to the remote application during periods of inactivity to detect TCP connection failures. The remote application must send a response back to the Xtended Services Platform. If the remote application fails to respond to consecutive handshake messages, then the Xtended Services Platform considers the connection as dead and releases the socket. When this happens, event channels created over the TCP connection are invalidated, which allows the Cisco BroadWorks core to initiate appropriate failover handling.

The remote application uses requests defined in this specification to publish and query its synchronized state to the Cisco BroadWorks core. For more information, see section [3.4.6 Application Controller \(AS only\)](#).

- At initialization, the remote application sends a request to publish its synchronized state to Cisco BroadWorks once it has synchronized with the state of Cisco BroadWorks resources.
- In steady mode of operation, the remote application sends a request to Cisco BroadWorks to query its synchronized state. This request is used as an audit to ensure that the remote application and Cisco BroadWorks are in agreement with the current state.

For information on high-availability configurations and constructs, see section [3.4 Events](#).

3.3 Requests

The remote application sends requests to Cisco BroadWorks to perform call and service control, and to initiate state synchronization through the creation of event subscriptions and event channels.

The following diagram provides a high-level view of the concepts related to requests.

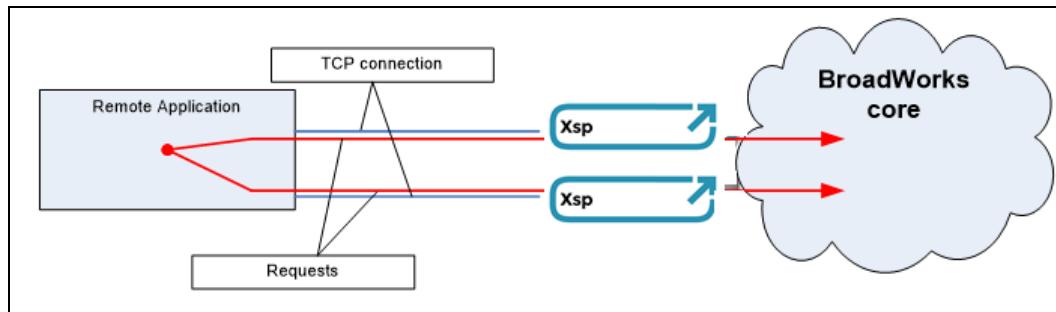


Figure 3 High-level View of Requests

In a high-availability deployment, the remote application connects to multiple Xtended Services Platform servers to ensure that the remote application maintains proper synchronization through event delivery. The remote application sends requests over these same connections.

Requests that are sent for the purpose of performing call and service control or for creating event subscriptions are typically load balanced across the TCP connections established to the different Xtended Services Platform servers.

Requests that are sent for the purpose of creating or managing event channels must be sent on the TCP connection that is relevant to that event channel.

For information on event channels and event subscriptions, see section [3.4 Events](#).



In all cases, Cisco BroadWorks performs authentication, processes the request, and then sends a response back to the remote application in the form of an XML payload.

3.3.1 Authentication Scheme and Session Management

Cisco BroadWorks authenticates requests received from the remote application using a basic authentication scheme. The remote application must include user credentials in all requests sent to Cisco BroadWorks. The user credentials are a base64-encoded string that consists of a userId and password separated by a colon. The userId may be that of a Cisco BroadWorks administrator (**AS only**) or of a Cisco BroadWorks subscriber. A Cisco BroadWorks administrator cannot be used to create a subscription for a user, group, or enterprise hosted on an Execution Server Telephony Application Server. When Cisco BroadWorks authenticates a request, it creates a session for the authenticated user and returns a session ID in the response. The session ID should be included in subsequent requests to ensure that user authentication is not performed on every single request.

This session times out after a configurable period of inactivity. When Cisco BroadWorks receives a request with an invalid session ID (for example, session has expired) or receives a request without a session ID, then the request is authenticated and a new session is created. In this case, the new session ID is returned in the response.

When Cisco BroadWorks fails to authenticate a request, then an authentication failure response is returned to the remote application.

The authenticated user defines the scope on which a given request may operate. In this specification, the authenticated user is distinguished from the entity targeted by a request. The authenticated user may or may not be the same as the subscriber targeted by a given request. In the case where they differ, scope rules are enforced to ensure that the authenticated user has the proper authorization to invoke a request on the target subscriber.

3.3.2 Trusted Client Application

It is possible to establish trusted client applications by supplying the Xsp with a TLS certificate that is used to identify the client. Trusted applications can establish system channels which can then be used to subscribe to multiple events using a client identity. The usage of the client identity allows subscribing without using Cisco BroadWorks users credentials. For more information on this mechanism, see the *Simplify Notifications to UCaaS and Collaborate Feature Description* [15].

3.4 Events

The remote application uses the events to maintain state synchronization with Cisco BroadWorks resources (for example, call state, ACD queue, and route point queue).

This section describes the concepts related to the delivery of events from the Cisco BroadWorks core to the remote application, principally the concepts of event subscriptions and event channels. From a high-level perspective, events are delivered from Cisco BroadWorks to the remote application in the context of event subscriptions over an event channel that was previously created.

The following diagram provides a high-level view of the concepts related to events.

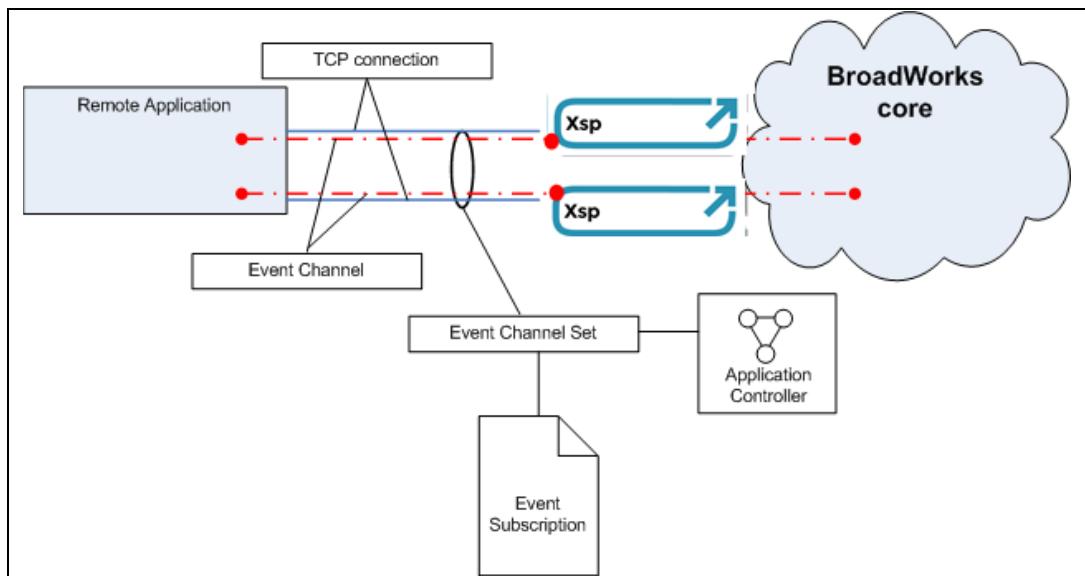


Figure 4 High-level View of Events

To initiate the delivery of events from the Cisco BroadWorks core, the remote application creates an event channel over a TCP connection and associates it to an event channel set. The remote application then creates an event subscription, which is associated with the same event channel set.

- **Event Channel Set:** An event channel set consists of one or more event channels.
- **Event Channel:** An event channel is a logical path between the remote application and the Cisco BroadWorks core. The event channel is created by the remote application and is bound to a specific TCP connection. The event channel is characterized by a priority and a weight, which allows the remote application to control event routing from the Cisco BroadWorks core, specifically when multiple event channels constitute the event channel set. For more information, see section [3.4.4 Event Channel](#).
- **Event Subscription:** An event subscription is used by the remote application to subscribe to specific event triggers in the Cisco BroadWorks core. An event subscription is associated with an event channel set. When an event is triggered on Cisco BroadWorks per the event subscription, the event is sent to the remote application over an event channel selected from that set. For more information, see section [3.4.5 Event Subscription](#).
- **Application Controller:** The application controller allows the remote application to publish its ready state to the Cisco BroadWorks core. The ready state is used by Cisco BroadWorks to evaluate failure conditions for specific functions (for example, for a route point). The controller is optional and is associated to an event channel set. The controller is managed through configuration in the Cisco BroadWorks core and has two states: “not ready”, “ready”. The remote application sets the controller state to ready once it has created event channels and subscriptions, and has synchronized its state with the Cisco BroadWorks core. For more information, see section [3.4.6 Application Controller \(AS only\)](#).

These constructs allow the remote application to function in high-availability mode. Depending on the level of protection required, a remote application establishes one or more TCP connections to one or more Xtended Services Platform servers, creates one or more event channels, and creates one or more event subscriptions.

The Xtended Services Platform sends a keepalive message to the remote application to detect TCP connection failures. The remote application must send a response back to the Xtended Services Platform. If the remote application fails to respond to consecutive keepalive messages, then the Xtended Services Platform considers the connection as dead and releases the socket. When this occurs, event channels created over the TCP connection are invalidated, which allows the Cisco BroadWorks core to initiate appropriate failover handling.

If the remote application sends a keepalive message, Cisco BroadWorks responds to it.

Cisco BroadWorks supports various configurations to support remote applications in a protected mode. The following subsections describe a sample of these configurations.

3.4.1 High-Availability Configuration One – Active-Standby with Synchronization

In this configuration, the remote application runs on two servers. The first server is actively performing service execution and events from Cisco BroadWorks are routed in priority to that server. The second server is in standby. It is synchronized with the first server and ready to take over at any time in case of failure.

To support this configuration, each server creates a single event channel set and a single event subscription. To enable protection at the server and connection levels, four event channels are created over distinct TCP connections.

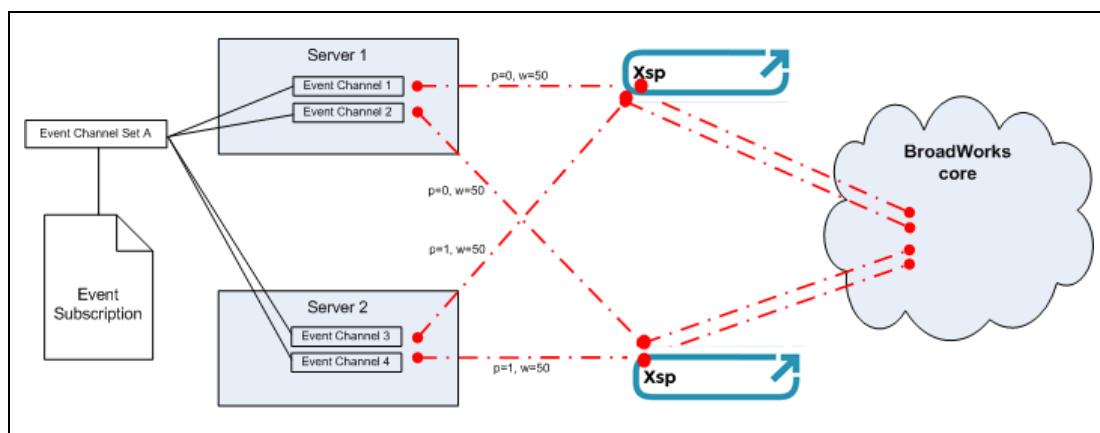


Figure 5 Active-Standby with Synchronization

All event channels are associated with the same event channel set A. Event channels 1 and 2 have the same priority and the same weight. Event channels 3 and 4 have the same priority and the same weight. The priority of event channels 1 and 2 is higher than the priority of event channels 3 and 4.

Events are triggered in the Cisco BroadWorks core and are sent to remote application according to the event subscription. The Cisco BroadWorks core selects a channel from the event channel set A according to the priority and weight parameters of the channels. Based on the priority and weight values shown in the diagram, events are delivered to the server 1 in priority and load balanced across the event channels 1 and 2. Events are only sent to the second server when these two event channels have failed.

3.4.2 High-Availability Configuration Two – Active-Active with Synchronization

In this configuration, the remote application runs on two or more servers. Service execution can be performed on either server and events from Cisco BroadWorks can be routed to either server. The two servers are synchronized and service execution is load balanced across the two servers.

To support this configuration, each server creates a single event channel set and a single event subscription. To enable protection at the server and connection levels, four event channels are created over distinct TCP connections.

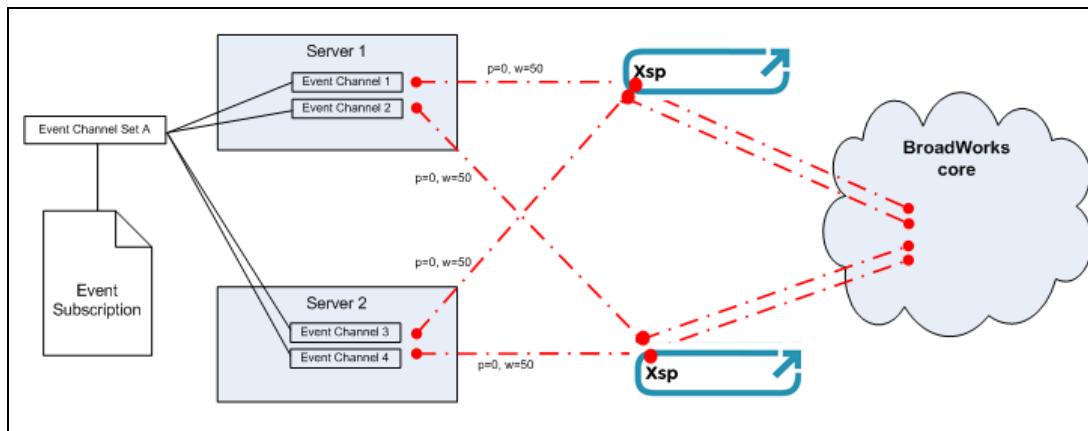


Figure 6 Active-Active with Synchronization

All event channels are associated with the same event channel set A, and have the same priority and the same weight.

Events that are triggered in the Cisco BroadWorks core are load balanced across the four event channels and can be processed by any of the two servers.

3.4.3 High-Availability Configuration Three – Active-Active Without Synchronization

In this configuration, the remote application runs on two or more servers. Service execution can be performed on either server, but the two servers are not synchronized in any way. From the perspective of Cisco BroadWorks, two distinct remote applications have created the same subscriptions. Events from Cisco BroadWorks are triggered twice (per each subscription) and routed to both servers to ensure that they both individually maintain sync with the Cisco BroadWorks core.

To support this configuration, each server creates two event channel sets and two event subscriptions. To enable protection at the server and connection levels, two event channels are created for each event channel set over distinct TCP connections.

This configuration is more intensive on network resources since the event subscriptions are duplicated. When an event triggers in the Cisco BroadWorks core, the same event needs to be instantiated and delivered to the remote application.

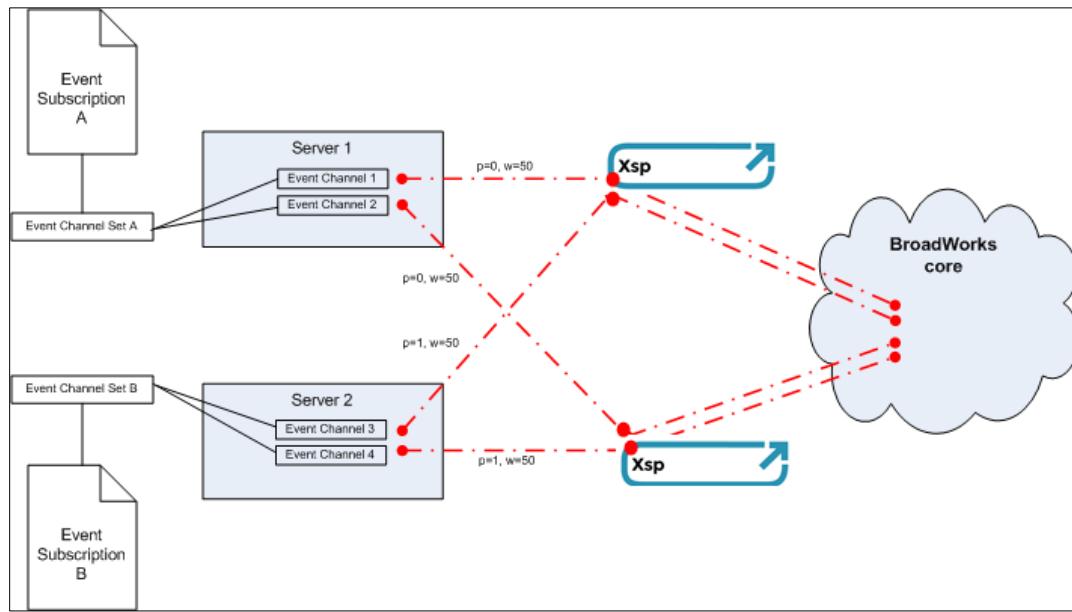


Figure 7 Active-Active Without Synchronization

Event channels 1 and 2 (server 1) are associated with event channel set A, and have the same priority and the same weight. Event channels 3 and 4 (server 2) are associated with event channel set B, and have the same priority and the same weight.

In this case, each server has its own event subscription. Event subscription A is associated with event channel set A and event subscription B is associated with event channel B.

Events are triggered in the Cisco BroadWorks core and are sent to remote application according to the two event subscriptions. The two event subscriptions are handled independently of one another. For the events triggered per event subscription A, the Cisco BroadWorks core selects a channel from the event channel set A. Based on the priority and weight values shown in the diagram, events are delivered to server 1 and load balanced across the event channels 1 and 2. For the events triggered per event subscription B, the Cisco BroadWorks core selects a channel from the event channel set B. Based on the priority and weight values shown in the diagram, events are delivered to server 2 and load balanced across the event channels 3 and 4.

3.4.4 Event Channel

The remote application creates event channels by sending an Add Channel request to Cisco BroadWorks over a specific TCP connection. This request contains the following parameters:

- A channel set identifier. This is any free-form text that identifies the event channel set to which this channel belongs. This identifier is meaningful in the context of an event subscription and must be globally unique for a subscriber. Uniqueness is necessary to avoid interactions with other applications that may be creating event subscriptions for the subscriber on distinct event channels.
- The priority and weight of the channel within the event channel set. This is used by Cisco BroadWorks when determining the event channel on which to deliver an event.
- An expiration value. The expiration value specifies the life duration of the event channel.
- An application ID (**AS only**). This optional parameter uniquely identifies a remote application and can be used as a mechanism to limit the number of instances of a remote application.

If the Add Channel request is accepted, then Cisco BroadWorks assigns a channel identifier to the event channel and returns it in the response.

An event channel is owned by the authorized subscriber that sent the Add Channel request. If the remote application creates multiple event channels with the same channel set identifier, then the same authorized subscriber must send all Add Channel requests associated with the channel set.

By default, Cisco BroadWorks allows a maximum of 8 channels per channel set. This maximum can be configured between 1 and 100. When this limit is exceeded, the channel replaces the channel that is closest to expiration.

3.4.4.1 Channel Requests

The following table provides a brief description of the requests that can be performed on an event channel.

Request	Description
Add Channel	Add an event channel. Returns a channelID identifying the channel.
Update Channel	Update an existing event channel. This request is used to extend the life of an event channel by modifying the channel expiration value.
Delete Channel	Delete an existing event channel.
Get Channel	Get a list of event channels associated with a channel set.

3.4.4.2 Channel Life Duration

A remote application must provide an expiration value when creating an event channel. This value specifies the number of seconds the event channel will be active. This value must be within the range configured in the system.

An event channel expires when the calculated expiration time (time of creation plus expires value) is reached. When the channel expires, the channel is removed and can no longer be used for the delivery of events and a *Channel Terminated* event is generated.

To keep a channel active and prevent its removal, a remote application must refresh the channel prior to its expiration. This can be done by sending the Update Channel request.

3.4.4.3 Application ID

A remote application may include an application ID when creating a channel to enforce a limit of one single channel set per application. If a second channel set is created with the same application ID, then Cisco BroadWorks terminates all channels associated with the first channel set (effectively terminating the channel set itself). This mechanism can be used to limit the number of instances of a remote application by specifying the same application ID in a channel creation request.

3.4.4.4 Channel Terminated Event

When a channel is removed, Cisco BroadWorks sends a *Channel Terminated* event across that channel, except at system startup at which time no event is sent when the channel is removed.

A channel is removed under any of the following conditions:

- The remote application explicitly removes the channel by sending the Delete Channel request.
- The channel expires.

- The TCP connection is detected as being down. For information on the handshake mechanism, see section [3.2 High-Availability Solution](#).
- The remote application returns a 404 “ChannelId not found” as a response to an event. For more information, see section [3.4.5.13 Delivery](#).
- The authorized subscriber is deleted from the system.
- A channel set has been created with the same application ID, thereby terminating all channels associated with this first channel set.

3.4.5 Event Subscription

The remote application sends an Add Subscription request to Cisco BroadWorks to create a subscription. The request contains the following attributes:

- A target. A target is typically a subscriber hosted in Cisco BroadWorks. The target can also be a collection of subscribers (that is, a group, service provider/enterprise, or the system) or a subcomponent of a user.
- A channel set identifier. This is any free-form text that identifies the event channel set over which the events must be sent. This corresponds to the value provided when creating an event channel.
- An applicationId. This is any free-form text that identifies a remote client application. By using different applicationId, more than one client application can “subscribe against” the same target for the same event. Normally a remote application provides the same applicationId for all subscriptions.
- An event package. The event package defines how the subscription behaves, including what types of notifications are sent and when. Each event corresponds to various types of notifications that are asynchronously sent by Cisco BroadWorks back to the subscribing client application.
- An expiration value. The expiration value specifies the life duration of the subscription.

If the Add Subscription request is accepted, then Cisco BroadWorks assigns a subscriptionId to the subscription and returns it in the response.

After sending this response, Cisco BroadWorks sends an event that allows the remote application to know of the initial state of the target.

Then at various points in the life of a subscription, Cisco BroadWorks sends events to the remote client application. These events are issued on the target of the subscription when events occur in Cisco BroadWorks.

This section provides the details on event subscription, creation, deletion, modification, and retrieval.

3.4.5.1 Subscription Requests

The following table provides a brief description of the request that can be sent for a subscription.

Request	Description
Add Subscription	Add a subscription against the specified target. Returns a <i>subscriptionId</i> identifying the subscription. Can produce an initial event.
Update Subscription	Modify an existing subscription. Can produce an initial event. This request is used to extend the life of an event subscription by modifying the subscription expiration value.
Delete Subscription	Delete an existing subscription.

Request	Description
Get Single Subscription	Get a subscription based on subscriptionId.
Get Subscription	Get one or multiple subscription that matches search criteria.

3.4.5.2 Subscription Life Duration

A remote application must provide an expiration value when creating a subscription. This value specifies the number of seconds the subscription will be active. This value must be within the range configured in the system. If not, Cisco BroadWorks will adjust the value to meet the range configuration and return the adjusted value in the response.

When a subscription expires, a *Subscription Terminated* event is generated.

To keep a subscription active and prevent its termination, a remote application must refresh the subscription prior to its expiration. This can be done through the *Update Subscription* request.

3.4.5.3 Subscription Terminated Event

Event notifications are generated throughout the life of a subscription. The events sent are specific to the various event packages. For more information, see section [3.4.5.8 Event Package](#).

There is however one event that can be sent specifically for a subscription: the *Subscription Terminated* event. This event is produced when a subscription is terminated.

A subscription is terminated under any of the following conditions:

- The remote application explicitly deletes the subscription through a *Subscription Delete* action.
- The subscription expires.
- The Privacy setting for the target subscriber changes.
- The authorized subscriber is deleted from the system.
- The target subscriber (or group or service provider/enterprise) is deleted from the system.
- The system configuration is changed resulting in deletion of subscription.
- An authorized subscriber is an agent/supervisor of a Route Point or ACD and has subscriptions against the Route Point or ACD but then is removed as an agent/supervisor for ACD. In this case, all subscriptions for the user against the Route Point or ACD are deleted.
- All event channels associated with the event channel set are removed. In other words, the associated event channel set has no active event channels for the delivery of events.

3.4.5.4 Subscription Scope Rules

The remote application authorized subscriber is only allowed to interact with subscriptions whose owner is within the authorized subscriber scope. In other words, a remote application can send requests (like *Modify Subscription*) for a subscription within the scope of its authorized subscriber.

The following table summarizes the subscription scope rules enforced.

Authorize Subscriber Type	Subscription Scope
Subscriber	Can interact with subscription created by the subscriber

Authorize Subscriber Type	Subscription Scope
Department/group administrator (AS only)	Can interact with subscription created by any subscriber or administrator within the administrator's group.
Service provider/enterprise administrator (AS only)	Can interact with subscription created by any subscriber or administrator in the administrator's service provider/enterprise.
Provisioning administrator (AS only)	Can interact with subscription created by any subscriber or administrator except the system administrator.
System administrator (AS only)	Can interact with subscription created.

3.4.5.5 Target Scope Rules

The remote application authorized subscriber is only allowed to specify a subscription target within the authorized subscriber scope.

The following table summarizes the target scope rules enforced.

Authorize Subscriber Type	Target Scope
Subscriber	The subscription's target may be any subscriber in the subscriber's group/enterprise including the subscribers themselves. The target cannot be collections of subscribers such as a group or enterprise. If the subscription's target is an ACD, then the subscriber may also be any agent or supervisor for the ACD. If the subscription's target is a Hunt Group, then the subscriber may also be any agent for the Hunt Group.
Department/group administrator (AS only)	The subscription's target may be any subscriber in the administrator's group or can be the administrator's group itself.
Service provider/enterprise administrator (AS only)	The subscription's target can be any subscriber in the administrator's service provider/enterprise, any group within the administrator's service provider/enterprise, or the administrator's service provider/enterprise itself.
System/provisioning administrator (AS only)	The subscription's target can be any subscriber in the system, any group in the system, any service provider/enterprise in the system, or the system itself.

3.4.5.6 Event Channel Set Scope Rules

The remote application authorized subscriber must be the owner of the event channel set.

3.4.5.7 Event Types

Several types of event can be generated by Cisco BroadWorks. The following table lists the available event types.

Event Types	Description
Call	A call event is an event that reports a change related to a single call
Conference	A conference is an event that reports a change related to a conference.
Service	A service event is an event that reports a change to a subscriber service.

Event Types	Description
Route Point (AS only)	A Route Point event is an event that reports a change to a call maintained in the Route Point queue.
ACD (AS only)	An ACD event is an event that reports a change to a call maintained in the ACD queue.
Agent (AS only)	An agent event is an event that reports a change to an agent.
Call Park (AS only)	A Call Park event is an event that reports on calls that are parked against a specified subscriber.
Meet-Me Conference (AS only)	A Meet-Me conference event is an event that reports change related to a Meet-Me conference.
Collaborate Room (AS only)	A Collaborate Room event is an event that reports changes related to an active Collaborate Room.
Route Point	A conference event is an event that reports a change to a call maintained in the Route Point queue.
ACD	An ACD event is an event that reports a change to a call maintained in the Route Point queue.
Agent	An agent event is an event that reports a change to an agent.

More detailed information on events is provided in section [6 Events](#).

3.4.5.8 Event Package

A remote application must provide an event package when creating a subscription. The event package defines how the subscription behaves, including what specific types of notifications are sent and when. The following table summarizes the details of each available event package. It also provides the name of the initial event sent by Cisco BroadWorks following a subscription. Initial events allow a remote application to synchronize with the current state of the target.

Event Package Name	Event Type Generated	Initial Event
Basic Call	Include only a subset of the available call events. For more information, see section 3.4.5.9 Call Event Packages .	Call Subscription
Standard Call	Include only a subset of the available call and conference events. For more information, see section 3.4.5.9 Call Event Packages .	Call Subscription
Advanced Call	All call and conference events	Call Subscription
Call Forwarding Always	Call Forwarding Always Event	Call Forwarding Always
Call Forwarding No Answer	Call Forwarding No Answer Event	Call Forwarding No Answer
Call Forwarding Busy	Call Forwarding Busy Event	Call Forwarding Busy
Do Not Disturb	Do Not Disturb Event	Do Not Disturb
Call Transfer	Call Transfer Event	Call Transfer Event
Call Waiting	Call Waiting Event	Call Waiting Event
Last Number Redial	Last Number Redial Event	Last Number Redial Event

Event Package Name	Event Type Generated	Initial Event
Music On Hold	Music On Hold Event	Music On Hold Event
N-Way Calling	N-Way Call Event	N-Way Call Event
Third-Party Voice Mail Support	Third-Party Voice Mail Support Event	Third-Party Voice Mail Support Event
Three-Way Calling	Three-Way Call Event	Three-Way Call Event
Voice Messaging	Voice Messaging Event	Voice Messaging Event
BroadWorks Anywhere	BroadWorks Anywhere Event	BroadWorks Anywhere Event
Sequential Ringing	Sequential Ring Event	Sequential Ring Event
Simultaneous Ringing Personal	Simultaneous Ring Personal Event	Simultaneous Ring Personal Event
Voice Mail Message Summary	Voice Mail Message Summary Event	Voice Mail Message Summary
Hoteling Guest (AS only)	Hoteling Guest Event	Hoteling Guest
Call Park (AS only)	Call Parked Against Event	Call Parked Against Event
Call Center Configuration (AS only)	ACD Configuration Event	ACD Configuration Subscription
Call Center Monitoring (AS only)	Call Center Monitoring Event	Call Center Monitoring Event
Call Center Agent Monitoring (AS only)	Call Center Agent Monitoring Event	Call Center Agent Monitoring Event
Remote Office (AS only)	Remote Office Event	Remote Office Event
Route Point Queue (AS only)	Route Point Event	Route Point Subscription
CommPilot Express (AS only)	CommPilot Express Event	CommPilot Express Event
Call Center Queue (AS only)	ACD Event	ACD Subscription
Call Center Agent (AS only)	Agent Event	Agent Subscription
Meet-Me Conference (AS only)	Meet-Me Conference Event	Meet-Me Conference Subscription
Collaborate Room (AS only)	Collaborate Room Event	Collaborate Room Subscription
Flexible Seating (AS only)	Flexible Seating Guest Event	Flexible Seating Guest Event

3.4.5.9 Call Event Packages

There are three call event packages available:

- Basic
- Standard
- Advanced



When subscribing to the Basic Call event package, a remote application only gets a subset of the available call events, namely:

- Call Originating
- Call Originated
- Call Collecting
- Call Received
- Call Answered
- Call Released
- Call Releasing
- Call Subscription
- Hook Status (**AS only**)

When subscribing to the Standard Call event package, a remote application gets a subset of the available call and conference events. The notifications for this package include all the events sent by the Basic Call Event package plus the following ones:

- Call Held
- Call Retrieved
- Call Transferred
- Call Barged In
- Call Park Retrieved
- Call Picked-Up
- Call Forwarded
- Call Updated
- Conference Started
- Conference Held
- Conference Retrieved
- Conference Released
- Conference Updated
- Call Monitored (**AS only**)
- Call Recording Starting (**AS only**)
- Call Recording Started (**AS only**)
- Call Recording Paused (**AS only**)
- Call Recording Resumed (**AS only**)
- Call Recording Stopped (**AS only**)
- Call Security Classification Updated (**AS only**)
- Call Client Session Info Updated (**AS only**)

Finally, when subscribing to the Advanced Call event package, a remote application gets all the available call and conference event.



In contrast with the Basic and Standard Call event packages, the Advanced Call event package triggers the *Call Redirected* event when a call is redirected by the user (for example, by the Call Forwarding Always service).

The other distinction with the Basic and Standard Call event packages is in the trigger of the *Call Received* event. In the case of this event package, the *Call Received* event is sent for every incoming call, even for calls that are forwarded or released before actually being presented to the user's device.

In the case of the Basic and Standard Call event packages, the *Call Received* event is not sent if the incoming call is processed by one of the user's terminating services.

3.4.5.10 Initial Event

An initial event is issued only for subscription against the smallest event package scope. Typically, this means that an initial event is issued for user subscriptions. If the subscription target is a collection of subscribers (for example, a group), then no initial event is issued.

Some event packages however support subscription against a subscriber subcomponent (see sections [6.11 Meet-Me Conference Events \(AS only\)](#) and [6.12 Collaborate Room Events \(AS only\)](#) for examples). In such a case, an initial event is only issued for subscription against the target subID, not for subscription against a subscriber, or any larger scope target (group, enterprise, system).

3.4.5.11 Event Delivery

The events are delivered over the event channels created by the remote application and associated with the event channel set provided with the event subscription.

As mentioned in previously, each event channel has a priority and a weight.

The priority is used to order channels. The priority values range from 1 through 100 with 1 as default. The lower value has the higher priority.

The weight is used for load balancing. Weight values range from 1 through 100 with 50 as default. In the case of weight, the higher value has the higher weight. The weight value is relevant only in relation to other weight values with the same priority. A higher weight value means that the event channel is more likely to be selected.

An event channel is owned by the authorized subscriber that has created the channel. It is associated to the connection over which it was created and to the subscription(s) referencing the same event channel set identifier.

The following subsections provide more details on how a channel is selected and used to control event delivery.

3.4.5.12 Channel Selection

When an event notification needs to be sent for a specific subscription, Cisco BroadWorks selects an event channel from the event channel set associated with this subscription and sends the event over the event channel. The event is delivered through the Xtended Services Platform server over the TCP connection associated with the event channel.

If there is more than one event channel for a subscription, then Cisco BroadWorks selects the channel with the highest priority. Amongst channels with the same priority, the weight is used for load balancing. The higher the weight, the more probable the channel will be used. Once the channel is selected, the event is sent over the event channel.



If the event delivery fails, then Cisco BroadWorks selects the next channel based on the priority and weight. The event is delivered again over the selected event channel. By default, up to eight event channels can be associated with an event channel set. This maximum value can be changed through system configuration.

3.4.5.13 Delivery

The events delivered by Cisco BroadWorks have event information in the event types as described in section [6 Events](#). Since Cisco BroadWorks does not guarantee in-sequence event delivery, it is possible that events may be issued out of order if there are two or more events triggered very close together. For this reason, a sequence number is provided with each event. The sequence number parameter in any event should be used by a remote application to collate events in the correct order.

The remote application that receives an event is required to send a response. The response contains a code and a reason. The following table lists the possible code and reasons available to a remote application.

Code	Reason	Description
200	Ok	The event was successfully received by the remote application.
404	ChannelId not found	The channelId in the event is unknown by the remote application. Cisco BroadWorks deletes the event channel and sends the event over the next event channel in the set.
481	SubscriptionId not found	The subscriptionId in the event notification is unknown to the remote application. Cisco BroadWorks deletes the subscription and drops the event.
500	Server Internal Error	The remote application encountered an unexpected condition that prevents it from handling the event. Cisco BroadWorks sends the event to the next event channel in the set.
400	<remote application reason>	The remote application could not process this specific event. Cisco BroadWorks drops the event.

Cisco BroadWorks is stateful in terms of event delivery for a given event channel. If no response is received from a remote application (or a response is received outside of the configured wait time), then the event channel is tagged as non-responsive and a new event channel is selected for delivery of the event. If no other event channel is found, then the event is dropped, then the non-responsive event channel is removed and the event subscription is deleted.

As long as the event channel is tagged as “non responsive”, it is treated with the least priority in the selection algorithm. The “non responsive” condition expires per system configuration, at which point the event channel is untagged and treated accordingly in the selection algorithm.



3.4.5.14 Shared Subscription (**AS only**)

Subscription sharing allows a remote application that has multiple subscriptions to the same target and event package to share a user-level subscription in Cisco BroadWorks. When shared subscriptions are used, only one notification will be sent for each state update for a particular target user and event package combination, regardless of the number of users subscribing to the same event package and target combination. This improves the performance and scalability of the solution as it reduces the volume of event notifications between Cisco BroadWorks and the remote application.

Shared subscriptions can be used for example when a remote application creates subscriptions on behalf of ACD supervisors, where one individual subscription to an ACD is created for each supervisor served by the remote application. Without shared subscriptions, one event is issued for each subscription. As a result, the volume of event notifications handled by the remote application is dependent on the number of subscriptions created and therefore on the number of supervisors. With shared subscriptions, the number of event notification becomes independent of the number of subscriptions and therefore of the number of supervisors served, making the solution more scalable.

To share subscriptions, each remote application must send independent subscription requests to Cisco BroadWorks with the same shared subscription ID (SSID). Cisco BroadWorks uses the SSID in the subscription to link and correlate the subscriptions, and it only sends a single notification for state updates to the target, regardless of the number of remote applications sharing the subscription.

3.4.5.14.1 *Initial Event Notifications*

When Cisco BroadWorks handles a subscription request, it sends out an initial notification for the subscription. When shared subscriptions are used, it only sends out one initial notification for the first subscription using the SSID. If another subscription request is processed with the same SSID, then an initial notification is not sent out.

3.4.5.14.2 *Subscription Refresh*

When Cisco BroadWorks handles a subscription update request to refresh the expiry for a subscription, it updates the subscription expiry and sends out an initial notification event to allow the client application to resynchronize its state. With shared subscriptions, the Application Server updates the expiry for the specified subscription and only sends one initial notification event, even if the subscription is linked to other subscriptions with the same SSID.

3.4.5.14.3 *Subscription Termination*

If a terminated subscription is linked to other subscriptions with the same SSID, the other subscriptions are not impacted.

3.4.5.14.4 *Sequence Number*

The sequence numbers for the events for subscriptions that are linked by an SSID are maintained and incremented on each notification.

3.4.6 Application Controller (**AS only**)

The purpose of the application controller is to allow the remote application to publish its ready state. As part of its startup sequence, the remote application synchronizes with the Cisco BroadWorks resources state through the creation of event channels and subscriptions. Once the remote application has synchronized with all required Cisco BroadWorks resources, the application publishes itself to as “ready”.



The application controller is associated to an event channel set through configuration. When Cisco BroadWorks detects that an event channel set is empty (that is, when all event channels have been removed), then the application controller state is changed to “*not ready*”.

Cisco BroadWorks uses the state of the application controller to evaluate failure conditions of specific function. For example, a route point resource that is linked through configuration to one or more application controllers is moved to a failed state when an application controller transitions to the “*not ready*” state and no other application controllers are in the “*ready*” state.

The remote application communicates its ready state by sending the Set Application Controller State request. The request contains the following attributes:

- An application controller identifier: This is any free-form text that identifies an application controller.
- A state: Either “*ready*” or “*not ready*”.

An application controller is defined at the system level and its state can only be set by an authorized system administrator.

3.4.6.1 Application Controller Requests

The following table provides a brief description of the requests that can be performed on an application controller.

Request	Description
Set Application Controller State	<p>Sets the application controller’s state.</p> <p>This request is typically invoked once the remote application has synchronized all subscribed resources related associated with an event channel set. The target state is “<i>ready</i>” in this case. Cisco BroadWorks will reject the state transition if the associated event channel set is empty.</p> <p>This request can also be used to force the application controller state to “<i>not ready</i>” and force evaluation of a failure condition on resources that are linked to the application controller.</p>
Get Application Controller State	<p>Gets the application controller’s state.</p> <p>This request is used by the remote application to synchronize with the application controller state maintained by Cisco BroadWorks. It may also be used as a keepalive mechanism.</p>

3.5 Initialization

The following time sequence diagram shows the interaction between the remote application and the Xtended Services Platform as the remote application is starting up. For simplicity, responses are not shown in the diagram.

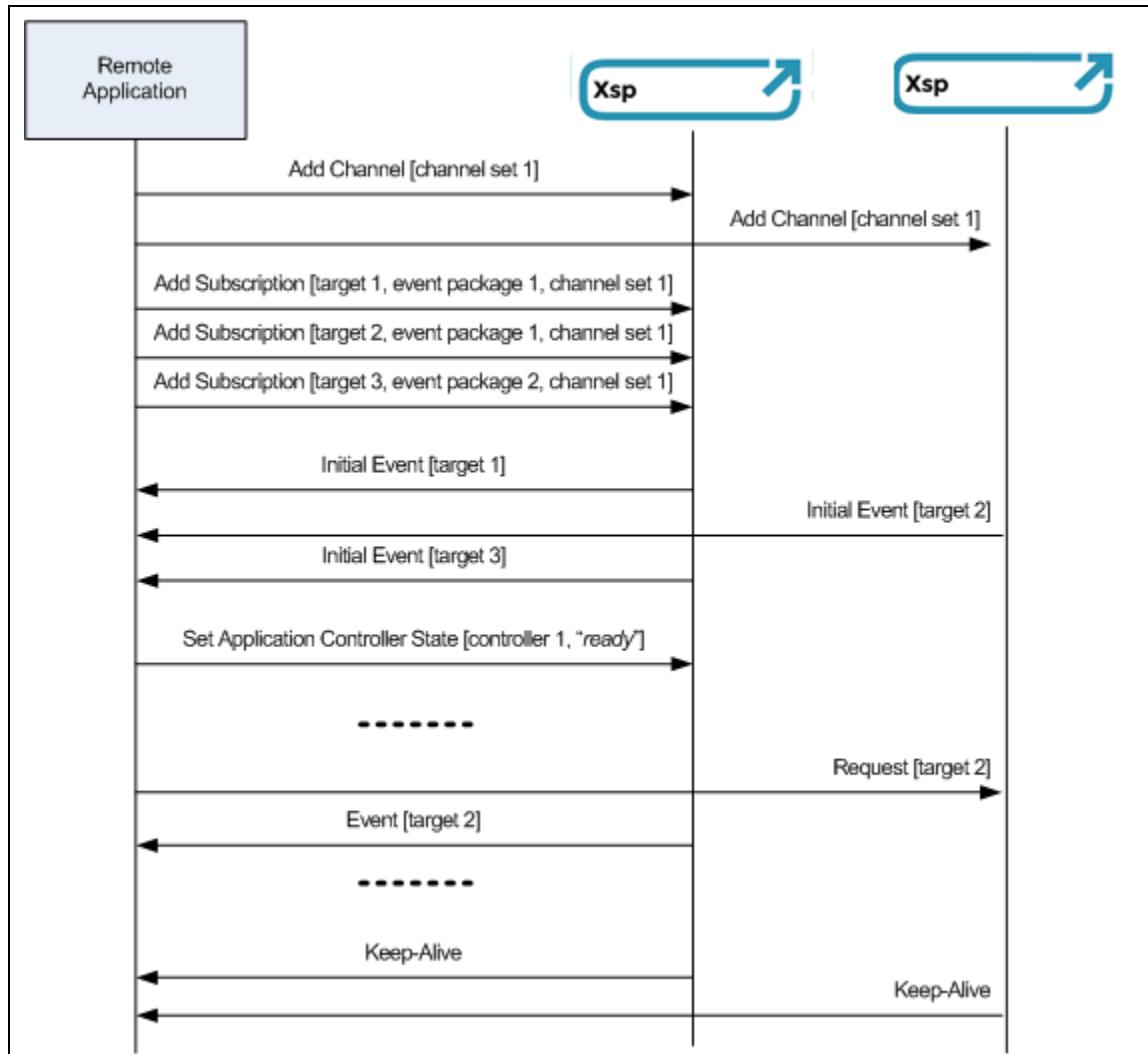


Figure 8 Initialization

- 1) The remote application establishes TCP connections to Xtended Services Platform servers and creates an event channel over each connection by sending the Add Channel request. In this example, the remote application creates two channels associated to the same event channel set 1.
- 2) The remote application creates a number of subscriptions depending on the resources that need to be synchronized with. The subscriptions make reference to the event channel set that contains the channels previously created. In this example, the remote application has subscriptions to an event package on two targets and a subscription for another event package on a different target. All subscriptions are associated with the same event channel set 1.



- 3) Cisco BroadWorks sends initial events for the three subscriptions created. The events allow the remote application to synchronize state with the target resources. In this example, one of the initial events is sent over the second event channel.
- 4) The remote application has finished synchronizing its state with Cisco BroadWorks and sends a Set Application Controller State request to change its state to “ready”.
- 5) The remote application is now running in steady state. The remote application sends requests to Cisco BroadWorks to trigger call and service control functions. Cisco BroadWorks sends events to the remote application. The Xtended Services Platform sends a keepalive message over each connection on a regular basis to detect connection failures. The remote application can also use the GetApplicationControllerState request as a keepalive mechanism.

3.6 Backward Compatibility

The interface defined in this specification is backward compatible for up to two major releases of Cisco BroadWorks beginning with Release 17.0 (Release Complete). This allows Cisco BroadWorks to be upgraded without requiring code changes to the remote application that interworks with Cisco BroadWorks.

3.6.1 Requests

Each request over the CTI is versioned. The remote application sending a request over the CTI interface specifies the request version by providing the version parameter as part of the request. Examples of request versions are “19.0”, “20.0”, and so on.

Each version maps to a specific request functionality. If any change occurs in a request parameter or in the parameters included in the response returned by Cisco BroadWorks, then a new version of the request is created and the existing version remains unchanged.

A remote application issuing a request providing a specific version always gets the same response format, for as long as that version of the request is supported by Cisco BroadWorks. A versioned request is supported for two major Cisco BroadWorks releases, after which issuing that version of the request results in a method not supported error response.

As an example, the version of the GetDoNotDisturbRequest described in the Release 20.0 CTI specification is “20.0”. Assuming a change is made to the request in BroadWorks Release 21.0 (for example, an additional parameter is returned in the response); this request could be issued with “20.0” or “21.0” provided in the version parameter. Specifying “20.0” against a BroadWorks Release 21.0 deployment returns a response identical to the one issued for a Release 20.0 deployment. Specifying “21.0” returns a response with the changes as documented for the command’s new version.

3.6.1.1 Error

For each request received, Cisco BroadWorks returns a response. When there is a failure, a response with appropriate error information is returned. Each error message contains an error code as well as summary text.

Future releases or maintenance patches will not change the meaning associated with the error code returned. However, future releases or maintenance patches can:

- Return new error codes for a request.
- Stop returning existing error codes for a request.
- Modify the summary text being returned for an error code since this text is for informational use only.

3.6.2 Events

Events in the CTI interface are delivered to the remote application when it subscribes for one/more event packages. Each event package in the CTI interface is versioned. The version of the event package is specified by the remote application via the AddUserSubscriptionRequest, AddGroupSubscriptionRequest, AddEnterpriseSubscriptionRequest, AddServiceProviderSubscriptionRequest, or AddSystemSubscriptionRequest. The version of the event package is determined by the version parameter provided with the request. Examples of even package versions are "19.0", "20.0", and so on.

Each version maps to a specific event package functionality. If a new event is to be sent as part of an event package, or if there is any change in the parameters for a specific event, then a new version of the event package is created and the existing version remains unchanged.

If a remote application subscribes for an event package with a given version, it is guaranteed to receive event notifications in a defined and unchanged format for as long as that event package version is supported, which is two major Cisco BroadWorks releases. If there is any change in an event delivered by Cisco BroadWorks, that changed event is only viewable by subscribing to the newer version of the applicable event package.

As an example, the "Advanced Call" event package version described in the Release 20.0 CTI specification is "20.0". Assuming that a new event is now delivered as part of the "Advanced Call" subscription on a BroadWorks Release 21.0 deployment, a subscription can be made with a version of "20.0" or "21.0". Creating an "Advanced Call" subscription version "20.0" on a BroadWorks Release 21.0 deployment would continue to deliver events identical to a BroadWorks Release 20.0 deployment. Creating an "Advanced Call" subscription version "21.0" would deliver events per the new version of the event package, including the new event.

3.6.3 Version Value

Request and event versions are defined using a character string. In general, this string corresponds to the major BroadWorks Release number (for example, "20.0") the request or event was introduced. However, in some cases, requests or events are made available earlier through the use of intermediate version number (for example, "18.sp3" or "19.sp5"). As a result, the request and event package versions can differ from the Cisco BroadWorks software release.

To illustrate this, assume that a new version of the GetDoNotDisturbRequest has been introduced in BroadWorks Release 20.0 and was made available in Release 19.0 through a software patch. This software patch provides access to this new functionality by introducing version "19.sp5" of the GetDoNotDisturbRequest. As a result, a remote application can make use of the new functionality by setting the version to "19.sp5" in the request, or could access the former functionality by leaving the version set to "19.0".

3.6.4 XML Schema

The XML schema defines types used by the CTI interface and is published with every release of Cisco BroadWorks. A schema is associated to a specific CTI interface version (that is, the schema associated with interface version "21.0" will be available with BroadWorks Release 21.0).

The XML schema can also be published if a software patches introduces a CTI interface change within a release (that is, the schema for interface version "19.sp5" is published along with its associated software patch).

The changes to the schema are documented with the schema files. As an example, assuming the DoNotDisturb type has been modified; schema files in which the modification occurred would document this change.



For more information on the XML schema, see section [7 XML Schema](#).

4 Call and Service Management Concepts

This section describes the core concepts required to understand how a remote application controls and monitors calls through Actions and Events. It covers the following topics:

- Call model
- Conference model
- Route Point (**AS only**)
- Automatic Call Distributor (**AS only**)
- Call Park (**AS only**)
- Meet-Me Conference (**AS only**)
- Collaborate Room (**AS only**)
- Service Management

4.1 Call Model

4.1.1 Subscriber

A subscriber is a user provisioned on Cisco BroadWorks. Each subscriber has a profile that defines call and service processing behavior. A subscriber has one or multiple communication devices, and uses these devices to communicate with other subscribers or with network parties.

4.1.2 Subscriber ID

The subscriber ID is an index that uniquely identifies a subscriber on Cisco BroadWorks. A remote application uses the *subscriberId* in messages to identify the subscriber on Cisco BroadWorks.

The subscriber ID can be in the form “user@domain”, (where domain must be included) or it can be an E.164 phone number.

Both formats can be used by a remote application to identify a subscriber in a request.

4.1.3 User ID

A user ID is a subscriber ID using the “user@domain” form. Cisco BroadWorks uses the “user@domain” format to identify subscriber in events and responses sent back to a remote application.

4.1.4 Address

An address can be a Session Initiation Protocol (SIP) URI, a tel URI, or a digit string (such as an extension).

In some cases, an address can be a directory number and in other cases, it can be destination digits.

A SIP URI starts with “sip:”. For a complete description of the SIP URI specification, see *RFC 3261*. The following is an example of a valid SIP URI:

- `sip:user@broadsoft.com`

A tel URI starts with “tel:”. For a complete description of the tel URI specification, see *RFC 3966*. The following are examples of tel URIs, which are considered valid:

- `tel:12403649000`

- tel:2403649001
- tel:9002

If an address does not start with “tel:” or “sip:”, then “sip:” or “tel:” is prefixed to the address by Cisco BroadWorks according to the following criteria:

- If an address contains all digits (including a potential leading “+” sign), then “tel:” is prefixed to it;
- Otherwise, “sip:” is added to the beginning.

4.1.4.1 Alternate Address

Cisco BroadWorks allows subscriber to have up to ten alternate addresses in addition to the subscriber’s main address. A subscriber can be reached through its main address but also through any of its alternate addresses.

Distinctive ringing patterns can be assigned to an alternate address. Hence, calls to an alternate address result in a distinctive ringing pattern that corresponds to the address that was used.

A distinctive call waiting tone can also be assigned to an alternate address. If a subscriber is busy (and Call Waiting has been enabled) an incoming call to the main address results in the usual call waiting tone, while an incoming call to any of the alternate numbers results in a distinctive call waiting tone.

4.1.5 Call Session

A call session is a set of subscribers logically connected through Cisco BroadWorks for the purpose of communicating. A call session is made of one or multiple calls with subscribers or network parties. A call session is stateless.

The following figure provides a graphical representation for a subscriber-to-subscriber call session.

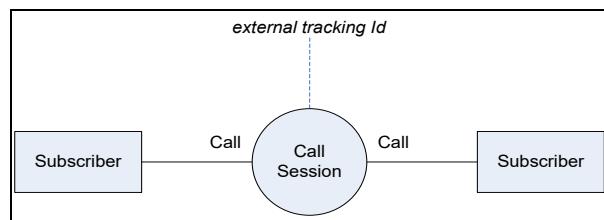


Figure 9 Call Session

4.1.5.1 External Tracking ID

An external tracking ID uniquely identifies a call session. The external tracking ID is included in Xtended Services Interface messages sent to the remote application and can be used to correlate various calls for a call session. The same external tracking ID is used for all calls logically connected to the call session. It is also used for the calls created in the context of redirections or forking. An external tracking ID is unique within an Application Server cluster.

4.1.6 Call

A call is a logical connection between a subscriber and a call session. A call is characterized by a call ID, call state, and a personality.

The following figure provides a graphical representation for a subscriber to network party call session. This graphical representation is also used in section [5 Requests](#) in describing the various requests. The connector between the subscriber and the call session represents a call, and the call state is overlaid on top of connector, as shown here.

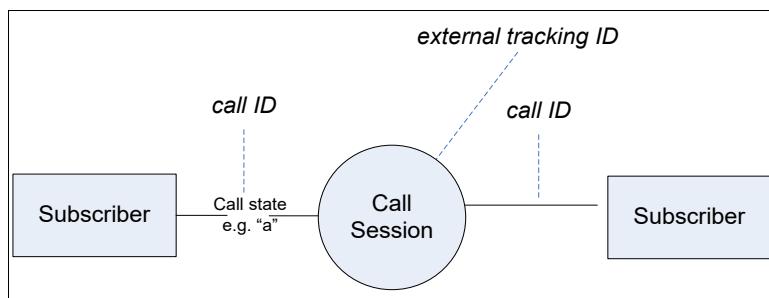


Figure 10 Subscriber to Network Party Call Session

Throughout its life, a call is connected to the same subscriber but can be connected to different call sessions.

4.1.6.1 Call ID

A call ID identifies a call. The call ID tracks a specific call within a call session. There is a different call ID for each call in a call session. A call ID is unique within an Application Server cluster.

4.1.6.2 Personality

The following table lists the three possible personalities of a call.

Personality	Description
Originator	Personality of the call when the subscriber is the initiator of the call session. If A initiates a call session to destination B, then the call associated to A has its personality set to "Originator".
Terminator	Personality of the call when the subscriber receives an incoming call. If A initiates a call to destination B, then the call associated to B has its personality set to "Terminator".
Click-to-Dial	Personality of the call when a "dial" request is being sent on behalf of a subscriber. For more information, see section 5.2.1 Dial . If a remote application tries to establish a call session from A to B (through a third-party call control dial request) then the call associated to A has its personality set to "Click-to-Dial" until A answers. Once A has answered, then a call attempt is made by Cisco BroadWorks to B and A's personality is set to "Originator".

4.1.6.3 Call State

A call can be in any of the following states. The abbreviated value is used in the diagrams contained in the following sections.

Call State	Abbreviated Value	Description
Alerting	a	This is the state in which an attempt is being made to establish communication. For example, when a caller manually dials a valid destination, the call is in the <i>Alerting</i> state while the communication is being established.
Active	A	The state in which a subscriber is actively participating in a call session. For example, when the remote subscriber goes off-hook on an incoming call, its call goes into the <i>Active</i> state.
Held	h	The state in which the remote party is inactively participating in a call session. This can happen for instance when the subscriber has put the call on hold.
Remote Held	rh	The state in which a subscriber is inactively participating in a call session. This can happen for instance when the subscriber has been put on hold.
Detached	d	The state in which the subscriber is no longer involved in the call session except for historical and tracking for example, billing purposes. This can happen for instance when an incoming call has been forwarded to another destination. Once the call has been forwarded, the forwarding subscriber is in the <i>Detached</i> state. The subscriber call no longer takes an Active role in the call session but is kept in the <i>Detached</i> state. Once in the <i>Detached</i> state, the call can only go into the <i>Released</i> state. Transition to other states is not possible.
Released	r	The state in which the subscriber is no longer involved in the call session. This can happen if for instance the subscriber goes on-hook.
Idle	i	The state in which there is no relationship between a subscriber and the call session.

4.1.6.4 Call Management Requests

The following table provides a brief description of the call related requests that can be performed on a subscriber. For more information, see section [5.2 Call Management Requests](#).

Request	Description
Dial	Initiate a call session from the specified subscriber to the specified destination.
Talk	Answer an unanswered call or retrieve a held call.
Hold	Put a call on hold.
Release	Release a call.
Blind Transfer	Transfer an existing call to a new destination in a single step.
Consultative Transfer	Transfer a specified call after consulting with the receiving party.
Mute Transfer	Transfer an existing call to a new destination in a single step. The transfer is considered successful if the transfer destination begins ringing.
Reconnect	Release a call and retrieve a held call in a single step.

Request	Description
Transfer to Voice Mail	Transfer an existing call to a voice mail.
Play DTMF	Send a series of digits.
Call Return	Initiate a call using the address of the last call missed or received.
Customer Originated Trace	Request to trace a call.
Directed Call Pickup	Pickup (answer) a call directed to another subscriber.
Directed Call Pickup with Barge-in	Pickup (answer) a call directed to another subscriber or barge in on the call if the call was already answered.
Last Number Redial	Originate a call session from the specified subscriber using the last address that was dialed.
Retrieve CallId List	Get a list of all the call IDs for the specified subscriber.
Retrieve Call Details	Retrieve the details of a call.
Monitor Call (AS only)	Silently monitor an existing call.
Monitor Next Call (AS only)	Silently monitor the next call received by an ACD, Route Point or silently monitor the next ACD or Route Point call answered by an agent.
FindMeFollowMe Call Push (AS only)	Push the call back to the Find-Me/Follow-Me group associated with the call.
Allocate IMRN	Allocate an IP Multimedia Routing Number.
Execute Assistant Call Push (AS only)	Push the call back to the executive associated with the call (see section 4.1.11 Executive Assistant (AS only) for details).
Execute Assistant Initiate Call (AS only)	Initiate a call session from the executive assistant on behalf of the executive (see section 4.1.11 Executive Assistant (AS only) for details).
Record (AS only)	Start recording a call.
Pause Recording (AS only)	Pause a call recording.
Resume Recording (AS only)	Resume call recording after a pause.
Stop Recording (AS only)	Stop recording a call.
Set Client Session Info (AS only)	Associate the client session info with the existing call session (see section 5.7.17 Set Conference Client Session Info).
Retrieve Hook Status (AS only)	Retrieve the hook status of a user.

4.1.6.5 Call Events

A series of call events can be generated against a call. Each event is associated to one or more call personalities. For instance, a *Call Originating* event can only be generated for calls having a “Click-To-Dial” personality. The following table provides a brief description of the existing call events. A complete description is provided in section [6.2 Call Events](#).

Events	Description	Call Personality
Call Originating	Event produced as the result of a Dial request.	Click-To-Dial
Call Collecting	Event produced when Cisco BroadWorks collects information by interacting with a subscriber. An example of this would be when a feature access code (FAC) is dialed.	Originator



Events	Description	Call Personality
Call Originated	Event produced when a call is originated by a subscriber.	Originator
Call Received	Event produced when a call is received by a subscriber.	Terminator
Call Answered	Event produced when a call is answered.	Originator and Terminator
Call Held	Event produced when a call is held.	Originator and Terminator
Call Retrieved	Event produced when a held call is retrieved.	Originator and Terminator
Call Updated	Event produced when the remote party information, local endpoint, or personality is changed.	Originator and Terminator
Call Redirected	Event produced when a call is redirected.	Originator and Terminator
Call Transferred	Event produced when a call is transferred by a remote party.	Originator and Terminator
Call Releasing	Event produced when a call is connecting to a treatment before being released.	Originator and Terminator
Call Released	Event produced when a call is released.	Originator and Terminator
Call Picked-up	Event produced when a call is picked-up.	Originator
Call Barged-in	Event produced when a barge-in occurs in a call.	Originator and Terminator
Call Monitored <i>(AS only)</i>	Event produced when a call is being monitored.	Originator and Terminator
Call Forwarded	Event produced when a call is forwarded as a result of a Cisco BroadWorks service (for example, Call Forward Always).	Originator
Call Detached	Event produced when a call is going to the detached state without being redirected.	Terminator
Call Subscription	Event produced on a new subscription to provide a description of all calls maintained by a subscriber.	Not applicable
Call Subscription Resync <i>(AS only)</i>	Event produced whenever the Application Server shuts down abnormally allowing a remote application to resynchronize.	Not applicable
Call Recording Starting <i>(AS only)</i>	Event produced when the call recording request has been processed but the recording cannot be started yet.	Originator and Terminator
Call Recording Started <i>(AS only)</i>	Event produced when call recording has started.	Originator and Terminator
Call Recording Paused <i>(AS only)</i>	Event produced when call recording is paused.	Originator and Terminator
Call Recording Resumed <i>(AS only)</i>	Event produced when call recording is resumed.	Originator and Terminator
Call Recording Stopped <i>(AS only)</i>	Event produced when call recording is stopped.	Originator and Terminator

Events	Description		Call Personality
Call Security Classification Updated (AS only)	Event produced when there are changes to the security classification characteristics of a call.		Originator and Terminator
Call Client Session Info Updated (AS only)	Event produced when there are changes to the client session info for the call.		Originator and Terminator
Hook Status (AS only)	Event produced when the hook status for a user is modified.		Originator and Terminator

4.1.6.6 Call State Tables

Call state behavior can be modeled using event driven finite state machines. Because the call state behavior is different for each call personality, there is one state machine per personality, resulting in a total of three state machines.

4.1.6.6.1 Originator State Table

The call state machine for call with the personality set to Originator is described using the following state transition table.

The table shows how the combination of the current state (as listed in the first row) with the occurrence of a specific event (as listed in the first column) can produce a state change. For instance, when a “Call Collecting” event occurs in the *Idle* state, the call state moves to *Active* (as shown at the intersection of the first row and first column). A “-” means that the event cannot be generated in the specified state.

Events	Current States						
	Idle	Alerting	Active	Held	Remote Held	Detached	Released
Call Originating	-	-	-	-	-	-	-
Call Collecting	Active	Active	-	-	-	-	-
Call Originated	Alerting	Alerting	Alerting	-	-	-	-
Call Received	-	-	-	-	-	-	-
Call Answered	-	Active	-	-	-	-	-
Call Held	-	Held	Held, Remote Held <i>See NOTE 1</i>	-	Held	-	-
Call Retrieved	-	-		Active	Active	-	-
Call Updated	-	Alerting	Active	Held	Remote Held	-	-
Call Redirected	-	-	Detached	Detached	Detached	-	-
Call Transferred	-	-	Alerting Remote Held Active <i>See NOTE 2</i>	Alerting Held <i>See NOTE 2</i>	Alerting Remote Held Active <i>See NOTE 2</i>	-	-
Call Releasing	-	Alerting				-	-



Events	Current States						
	Idle	Alerting	Active	Held	Remote Held	Detached	Released
Call Released	–	Released	Released	Released	Released	Released	Released
Call Picked-up	–	Active	–	–	–	–	–
Call Barged-in	–	–	Active	Held	Active	–	–
Call Monitored <i>(AS only)</i>	–	–	Active	Held	Active	–	–
Call Forwarded	–	Alerting	–	–	–	–	–
Call Recording Starting <i>(AS only)</i>	–	–	Active	Held	Remote Held	–	–
Call Recording Started <i>(AS only)</i>	–	–	Active	Held	Remote Held	–	–
Call Recording Paused <i>(AS only)</i>	–	–	Active	Held	Remote Held	–	–
Call Recording Resumed <i>(AS only)</i>	–	–	Active	Held	Remote Held	–	–
Call Recording Stopped <i>(AS only)</i>	–	–	Active	Held	Remote Held	–	–
Call Security Classification Updated <i>(AS only)</i>	–	Alerting	Active	Held	Remote Held	–	–
Call Client Session Info Updated <i>(AS only)</i>	–	Alerting	Active	Held	Remote Held	–	–
Hook Status <i>(AS only)</i>	–	–	–	–	–	–	–

NOTE 1: If the Hold was performed by the subscriber then the call goes into *Held* state. If the Hold was performed by the remote party, then the call goes into *Remote Held*.

NOTE 2: The Call Transferred event can be generated following a transfer or a consultative transfer. The resulting state is governed by the mechanism defined by the various requests.



4.1.6.6.2 Terminator State Table

The call state machine for a call with a Terminator personality is described using the following state transition table.

Events	Current States						
	Idle	Alerting	Active	Held	Remote Held	Detached	Released
Call Originating	–	–	–	–	–	–	–
Call Collecting	–	–	–	–	–	–	–
Call Originated	–	–	–	–	–	–	–
Call Received	Alerting	–	–	–	–	–	–
Call Answered	–	Active	–	–	–	–	–
Call Held	–	Held	Held, Remote Held See NOTE 1	–	Held	–	–
Call Retrieved	–	–	–	Active	Active	–	–
Call Updated	–	Alerting	Active	Held	Remote Held	–	–
Call Redirected	–	Detached	Detached	Detached	Detached	–	–
Call Transferred	–	Alerting	Alerting Remote HeldActive See NOTE 2	Alerting Held See NOTE 2	Alerting Remote HeldActive See NOTE 2	–	–
Call Releasing	–	–	–	–	–	–	–
Call Released	–	Released	Released	Released	Released	Released	Released
Call Picked-up	–	–	–	–	–	–	–
Call Barged-in	–	–	Active	Held	Active	–	–
Call Monitored <i>(AS only)</i>	–	–	Active	Held	Active	–	–
Call Forwarded	–	–	–	–	–	–	–
Call Detached	–	Detached	–	–	–	–	–
Call Recording Starting <i>(AS only)</i>	–	–	Active	Held	Remote Held	–	–
Call Recording Started <i>(AS only)</i>	–	–	Active	Held	Remote Held	–	–
Call Recording Paused <i>(AS only)</i>	–	–	Active	Held	Remote Held	–	–
Call Recording Resumed <i>(AS only)</i>	–	–	Active	Held	Remote Held	–	–

Events	Current States						
	Idle	Alerting	Active	Held	Remote Held	Detached	Released
Call Recording Stopped (AS only)	–	–	Active	Held	Remote Held	–	–
Call Security Classification Updated (AS only)	–	Alerting	Active	Held	Remote Held	–	–
Call Client Session Info Updated (AS only)	–	Alerting	Active	Held	Remote Held	–	–
Hook Status (AS only)	–	–	–	–	–	–	–

NOTE 1: If the hold was performed by the subscriber then the call goes into the *Held* state. If the hold was performed by the remote party, then the call goes into *Remote Held*.

NOTE 2: The Call Transferred event can be generated following a transfer or a consultative transfer. The resulting state is governed by the mechanism defined by the various requests. For more information, see section [5.2 Call Management Requests](#).

4.1.6.6.3 Click-to-Dial State Table

The call state machine for call with a Click-to-Dial personality is described using the following state transition table.

Events	Current States						
	Idle	Alerting	Active	Held	Remote Held	Detached	Released
Call Originating	Alerting	–	–	–	–	–	–
Call Collecting	–	–	–	–	–	–	–
Call Originated	–	–	–	–	–	–	–
Call Received	–	–	–	–	–	–	–
Call Answered	–	–	–	–	–	–	–
Call Held	–	–	–	–	–	–	–
Call Retrieved	–	–	–	–	–	–	–
Call Updated	–	Alerting	–	–	–	–	–
Call Redirected	–	–	–	–	–	–	–
Call Transferred	–	–	–	–	–	–	–
Call Releasing	–	–	–	–	–	–	–
Call Released	–	Released	–	–	–	–	–

Events	Current States						
	Idle	Alerting	Active	Held	Remote Held	Detached	Released
Call Picked-up	–	–	–	–	–	–	–
Call Barged-in	–	–	–	–	–	–	–
Call Monitored <i>(AS only)</i>	–	–	–	–	–	–	–
Call Forwarded	–	–	–	–	–	–	–
Call Recording Starting <i>(AS only)</i>	–	–	–	–	–	–	–
Call Recording Started <i>(AS only)</i>	–	–	–	–	–	–	–
Call Recording Paused <i>(AS only)</i>	–	–	–	–	–	–	–
Call Recording Resumed <i>(AS only)</i>	–	–	–	–	–	–	–
Call Recording Stopped <i>(AS only)</i>	–	–	–	–	–	–	–
Call Security Classification Updated <i>(AS only)</i>	–	Alerting	Active	Held	Remote Held	–	–
Call Client Session Info Updated <i>(AS only)</i>	–	–	–	–	–	–	–
Hook Status <i>(AS only)</i>	–	–	–	–	–	–	–

4.1.6.7 External Tracking ID Change

In some cases, an existing call can be connected to a new call session. As a result, the external tracking ID associated with the call can change. This type of change occurs when any of the following functionalities are used:

- Transfer with Consultation
- Mute Transfer
- Directed Call Pickup
- Directed Call Pickup with Barge-In
- Monitor Call
- Monitor Next Call

These functionalities can be exercised through a remote application request or directly from the phone interface.

In all cases, the external tracking ID changes are reported to a remote application by one of the following events:

- Call Transferred
- Call Picked-up
- Call Barged-in
- Call Monitored
- Call Updated
- Route Point Call Updated
- ACD Call Updated

Section [5 Requests](#) describes how each request impacts the external tracking ID and explains how a remote application can map or manage the change in the ID.

4.1.7 Endpoint

An endpoint represents an addressable location that terminates a call.

4.1.7.1 Access Endpoint

An access endpoint is an endpoint provisioned in Cisco BroadWorks to be associated with a subscriber. An access endpoint is identified using its address of record.

A SIP URI identifying a subscriber SIP phone is an example of an access endpoint as shown in the following figure.

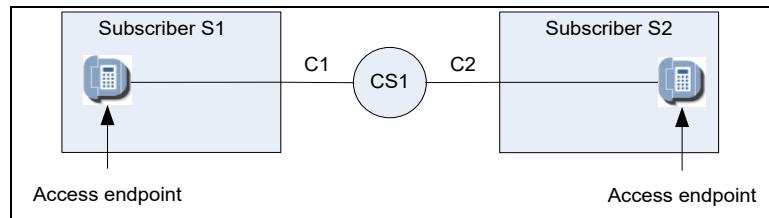


Figure 11 Access Endpoint

An access endpoint can change throughout the duration of a call. For instance when a subscriber performs a Call Retrieve from a phone, then the call access endpoint is expected to be modified. For more information on the Call Retrieve service, see the *Cisco BroadWorks Shared Call Appearance Interface Specification* [4].

4.1.7.1.1 Subscriber with Multiple Access Endpoints

Subscriber can also have multiple endpoints assigned. This can happen for instance when a subscriber has been provisioned to have incoming calls appear at multiple locations simultaneously (for example, when using Shared Call Appearance service). All endpoints where call appearance is shared can be used to answer the incoming call. In such a case, one of the endpoints is configured as the primary endpoint and the other endpoints are considered alternate endpoints. For more information on the Shared Call Appearance service, see the *Cisco BroadWorks Shared Call Appearance Interface Specification* [4].

4.1.7.2 Service Endpoint

Some services (for example, Executive service) allow a user to have their access endpoint provisioned as another user location. This can be used for originating calls from this alternate location, or for terminating calls to this location.

4.1.8 Party

A party is a participant in a call session. A party does not have to be a Cisco BroadWorks subscriber.

4.1.8.1 Network Party

A network party is a party participating in call session that is either *not* a Cisco BroadWorks subscriber or a Cisco BroadWorks subscriber who is *not* in the same group or enterprise.

Since a call is a logical connection between a subscriber and a call session, a connection between a network party and a call session is not a call, as shown in the following diagram.

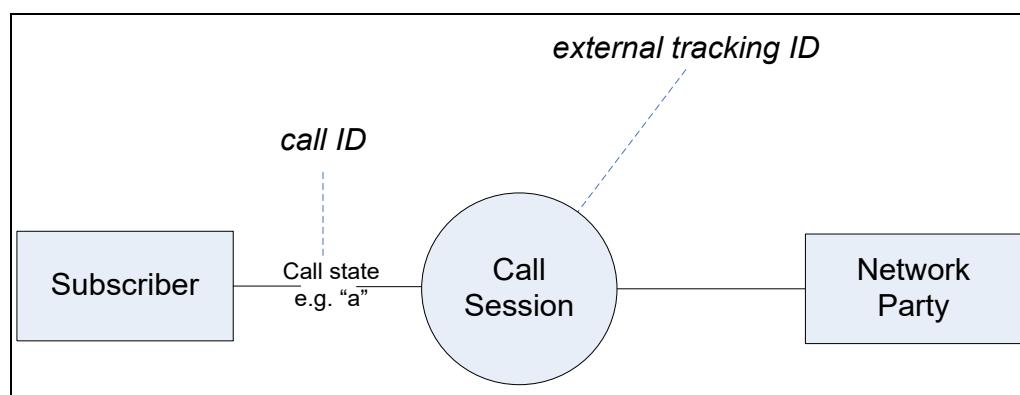


Figure 12 Network Party

Because there is no call between the network party and a call session, it is not possible to send requests or receive events from a network party connection to a call session. Although Cisco BroadWorks implements a state machine for the connection between the call session and the network party, the network party cannot be monitored nor can it be directly controlled.

4.1.9 Topologies

Depending on the locations of the party, multiple types of call session topologies are derived as described in the following subsections.

4.1.9.1 Intragroup/Enterprise Topology

If both parties of a call session are in the same group or enterprise, then a single call session with two calls is used to establish a communication between subscriber S1 and subscriber S2 as shown in the following diagram.

In the diagram, C1 represents the call between S1 and CS1 and C2 represents the call between S2 and CS1. CS1 represents the call session and is identified using its external tracking ID.

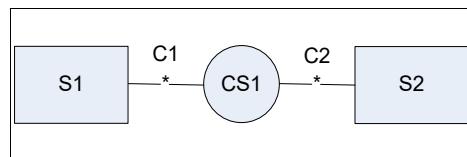


Figure 13 Intragroup/Enterprise Topology

4.1.9.2 Intra-Cisco BroadWorks Topology

If two parties in a call session (S1 and S2) are not in the same group or enterprise but are both Cisco BroadWorks subscriber, then two call sessions are required to establish a communication between them. The dashed line in the diagram simply illustrates that N1 and N2 exchange signaling information.

In this diagram, CS1 and CS2 are identified with their own unique external tracking ID.

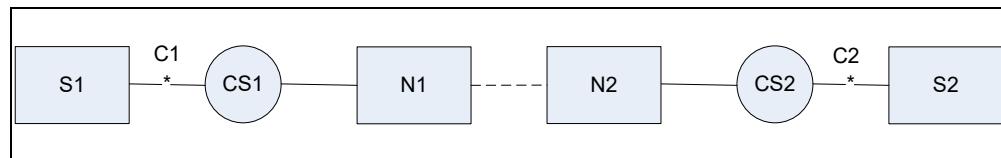


Figure 14 Intra-Cisco BroadWorks Topology

4.1.9.2.1 Redirections

In the previous diagram, if subscriber S2 redirects the call to a subscriber that is part of S1 group or enterprise, Cisco BroadWorks creates a new call CS3 for subscriber S3.

In this diagram, CS1, CS2, and CS3 are identified with their own unique external tracking ID.

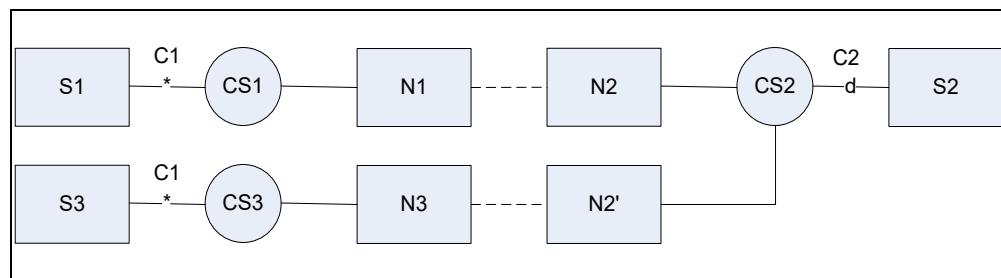


Figure 15 Redirections

4.1.9.3 Extra-Cisco BroadWorks Topology

If one of the two parties is not a Cisco BroadWorks subscriber, then a single call session is required to connect the Cisco BroadWorks subscriber (S1) to the network party (N1) as shown in the following diagram.

CS1 is identified with its external tracking ID.

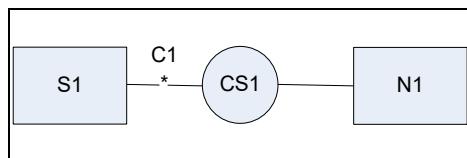


Figure 16 Extra-Cisco BroadWorks Topology

4.1.9.4 Single Call Topology

In some cases, only a single subscriber is connected to a call session. This can happen when Cisco BroadWorks is collecting information prior to calling a second subscriber.

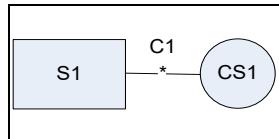


Figure 17 Single Call Topology

4.1.9.5 Call Types

Using the described topologies, various call types are defined as shown in the following table.

Call Type	Description	Call Session Topology
Group	The call session is between parties belonging to the same Cisco BroadWorks group.	Intra-group/enterprise
Enterprise	The call session is between parties belonging to the same Cisco BroadWorks enterprise.	Intra-group/enterprise
Network	One of the parties connected to the call session is not a Cisco BroadWorks subscriber or the two parties are not in the same group or enterprise.	Extra-Cisco BroadWorks and intra-Cisco BroadWorks topologies
Emergency	The call session is connected to an emergency party (for example, 911).	Extra-Cisco BroadWorks topology
Repair	The call session is connected to a repair party.	Extra-Cisco BroadWorks topology
Private Dial Plan	The call session was established by using a Private Dial Plan configured in Cisco BroadWorks.	Extra-Cisco BroadWorks topology
City-Wide Centrex	The call session was established by using a Private Dial Plan configured in Cisco BroadWorks for a City-Wide Centrex subscriber.	Extra-Cisco BroadWorks topology

4.1.10 Redirection

Some services allow incoming calls to be redirected.

The following diagram illustrates a simple case where subscriber S1 tries to contact subscriber S2 but subscriber S2 has redirected the call to subscriber S3.

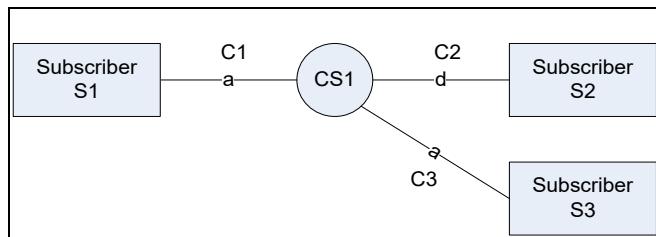


Figure 18 Call Session Redirection

In this diagram, both calls C2 and C3 contain information regarding the redirection performed. In addition, as shown, call C2 is in the *Detached* state following the redirection.

4.1.10.1 Call Forking

Some services, like Simultaneous Ringing, allow incoming calls to be forked.

The following diagram illustrates a simple case where subscriber S1 tries to contact subscriber S2 but a call toward subscriber S3 is attempted at the same time.

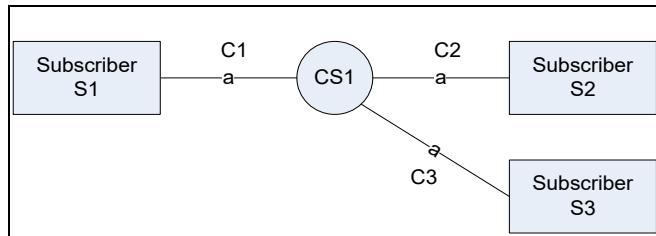


Figure 19 Call Session Forked

In a call forking scenario, calls C2 and C3 are in the *Alerting* state. This is different from the call redirection scenario previously shown where call C2 was in the *Detached* state (since it was no longer an active participant in the call session).

4.1.11 Executive Assistant (AS only)

The Executive and Executive-Assistant services provide a solution for executive/assistant interworking by enabling the following functionalities:

- **Executive Call Filtering:** This is the process of determining if an incoming call to an executive is to be filtered by the Executive service and hence should be routed to the executive's assistant.
- **Executive Call Screening:** After Executive Call Filtering has determined that an incoming call is to be filtered, the call is routed to the executive's assistant. When the Call Screening is disabled, the executive's own locations are not alerted for the filtered call. When Call Screening is enabled, the executive's own locations can also be alerted for the filtered call.
- **Executive-Assistant Call Initiation:** An assistant has the ability to initiate a call on behalf of an executive for which they are an assistant.

- Executive-Assistant Call Push: An assistant has the ability to push a call to the executive to which the call is associated.

4.1.11.1 Topology

The following diagram illustrates the resulting topology in which subscriber S1 (the assistant) has initiated a call toward subscriber S3 on behalf of subscriber S2 (the executive). All three calls, C1, C2, and C3, are connected through the same call session (CS1).

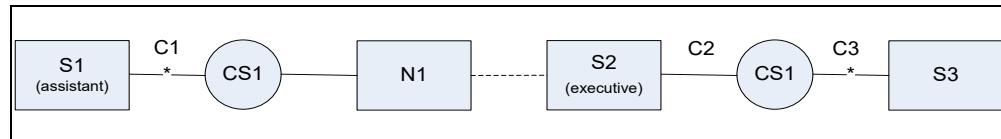


Figure 20 Executive Call Topology

When Subscriber S1 initiates call C1, call C2 is created. All call events generated for C1 (Call Originated, Call Answered, and so on) are also generated for call C2. If call C2 is released, call C2 and C3 are also released.

The same topology is also produced when a terminating call, filtered by the Executive service, is routed to the executive's assistant.

4.2 Conference Model

A conference is a set of call sessions logically connected by a subscriber called a controller. Each party connected to a call session of a conference is a participant in the conference. The conference party can actively communicate in the conference. A conference is identified by the controller subscriber ID and the controller call IDs connected to the conference.

The following figure provides a graphical representation of a conference where subscriber S1 is the controller and subscribers S2 and S3 are simple parties.

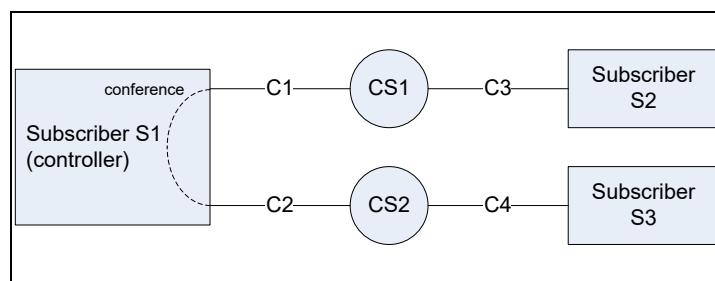


Figure 21 Conference

4.2.1 Conference States

The following table lists the possible conference states. The abbreviated values are used in the request descriptions in section [5 Requests](#).

State	Abbreviated Value	Description
Idle	i	The state in which there is no relationship between the call sessions and the conference.
Active	A	The state in which the controller is actively participating in a conference. For example, when the conference has been created, it goes into the Active state.

State	Abbreviated Value	Description
Held	h	The state in which the conference is put on Hold. This happens when the remote application performs a Conference Hold or when put on hold from the controller's endpoint. This is not the same has putting a specific call on Hold.
Released	r	The state in which the controller is no longer involved in the conference and participants can no longer use the conference to communicate. This can happen for instance if the controller goes on-hook.

4.2.2 Conference Requests

The following table provides a brief description of the requests that can be performed on a conference. A complete description for each request is provided in section [5.2.20 Call Me Now \(AS only\)](#).

Request	Description
Conference Start	Initiate a conference.
Conference Release	Terminate an existing conference.
Conference Hold	Put a conference on hold.
Conference Retrieve	Resume a held conference.
Conference Add Participant	Add a specified call as a participant to an existing conference.
Mute Call	Mute a call participating in a conference.
Unmute Call	Unmute a call participating in a conference.
Deaf Call	Make a call "deaf" participating in a conference.
Undeaf Call	"Undeaf" a call participating in a conference.

4.2.3 Conference Events

A series of conference events can be generated against a conference. The following table provides a brief description of the existing conference events. A complete description for each event is provided in section [6.3 Conference Events](#).

Events	Description
Conference Started	Event produced when a conference is started.
Conference Held	Event produced when a conference is held.
Conference Retrieved	Event produced when a conference is retrieved.
Conference Updated	Event produced when conference information is updated (for example, a participant is added).
Conference Released	Event produced when a conference is released.

4.2.4 Conference State Table

The conference state behavior can be modeled using an event-driven finite state machine. This state machine is best described using the following state transition table.

The table shows how the combination of the current state (as listed in the first row) with the occurrence of a specific event (as listed in the first column) can produce a state change. For example, when a “Conference Started” event occurs in the *Idle* state, the conference state moves to *Active* (as shown at the intersection of the first row and first column).

Events	Current States			
	Idle	Active	Held	Released
Conference Started	Active	–	–	–
Conference Held	–	Held	–	–
Conference Retrieved	–	–	Active	–
Conference Updated	–	Active	Held	–
Conference Released	–	Released	Released	–

4.2.5 Minimum Number of Participants in Conference

Two types of conference can be established as follows:

- Endpoint-initiated Cisco BroadWorks conference – This type of conference is established when a conference is started from the endpoint using the ad hoc conferencing methods defined in section 5.4 of the *Session Initiation Protocol (SIP) Call Control – Conferencing for User Agents RFC 4579* [10]. This conference can have zero or more participants associated.
- Cisco BroadWorks conference – This type of conference is established when a conference is started by a remote application request. This conference must have a minimum of two participants associated.

4.3 Route Point (AS only)

A Route Point is a virtual subscriber that is used by the remote application to queue and route calls. A Route Point is identified by its *subscriberId*. A Route Point maintains a queue of calls to be distributed.

4.3.1 Incoming Call Processing

The following figure shows how an incoming call is added to the Route Point queue as well as when it can leave the queue.

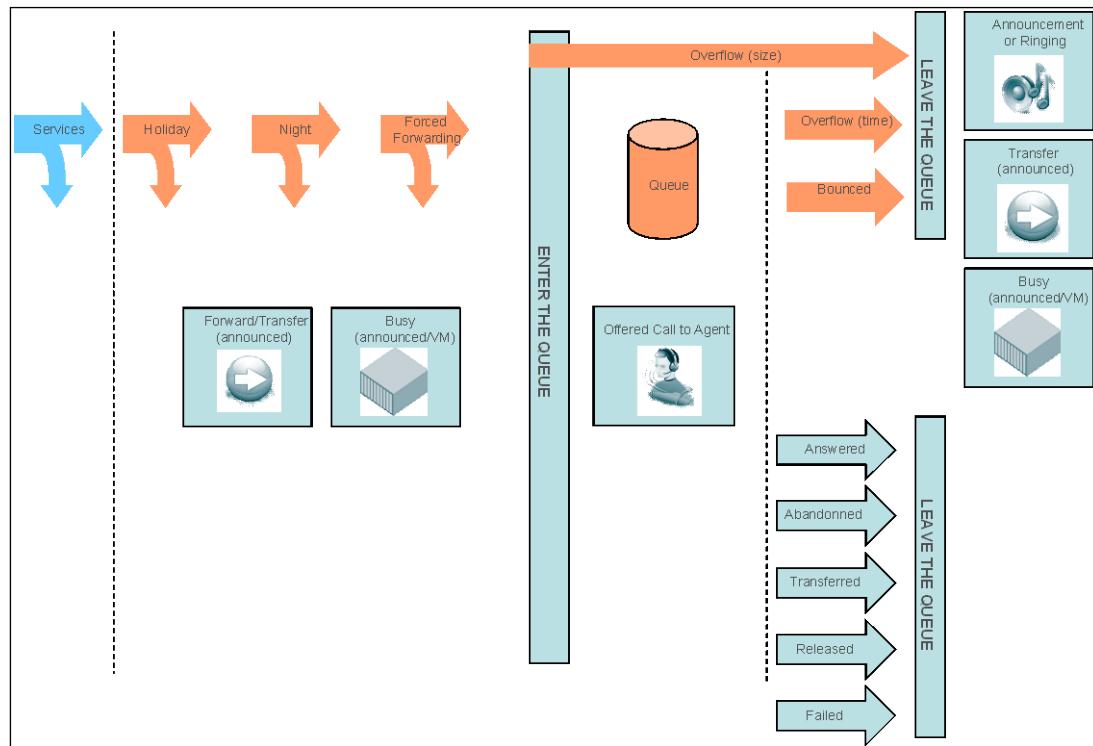


Figure 22 Incoming Call Processing

The processing of an incoming call to a Route Point can be summarized as follows:

An incoming call received by the Route Point is initially processed by the services that are assigned to the Route Point. In some cases, the services can prevent the call from proceeding to the queue. For example, the Call Forwarding Always service can be assigned and enabled for the Route Point, in which case all incoming calls are forwarded before entering the queue.

- The incoming call is then processed against the following Route Point policies:
 - The Holiday Service policy allows the configuration of a holiday schedule as well as the configuration of specific routing behavior for calls received during a holiday.
 - The Night Service policy allows the configuration of a business hour schedule as well as the configuration of specific routing behavior for calls received during non-business hours.
 - The Forced Forwarding policy allows the temporary diversion of new incoming calls to a new destination and is typically used when a change in business conditions requires calls to be redirected (usually to another Route Point) other than the Night Service or Overflow Routes.

- Note that a Route Point can have zero, one, or all of these policy assigned depending on the configuration.
- The incoming call enters the queue, unless the call was handled by a service or Route Point policy. If the queue has reached its maximum capacity, then the call is processed according to the Overflow policy resulting in a *Route Point Call Overflowed* event. Otherwise, the call is put into the queue and a *Route Point Call Added* event is generated.
 - The remote application can send various requests for the queued call as follows: Distribute the call to agent, play media, detect Dual-Tone Multi-Frequency (DTMF), and so on. For more information, see section [4.3.2 Route Point Requests](#).
- A call exits the queue under one of the following conditions:
 - An agent is offered the call and answers it, in which case a *Route Point Call Answered by Agent* event is generated.
 - The caller abandons the call by hanging up, in which case a *Route Point Call Abandoned* event is generated.
 - The remote application transfers the call out of the queue, in which case a *Route Point Call Transferred* event is generated.
 - The remote application releases the call, in which case a *Route Point Call Released* event is generated.
 - The call was not handled (either by an agent or another policy) within a specific amount of time, in which case a *Route Point Call Overflowed* event is generated.
 - A call is marked as *Failed*, in which case a *Route Point Call Failed* event is generated.

A call can also exit from the queue if it bounces and the Route Point Bounced Call policy is configured accordingly. In this case, a *Route Point Call Bounced* event is generated.

4.3.1.1 Bounced Call Policy

A call is bounced under any of the following conditions:

- When it is offered to an agent and the agent is not available for one of the following reasons:
 - Agent has no endpoint registered with Cisco BroadWorks
 - Agent has activated the Do Not Disturb service
 - Agent has explicitly rejected the offered call
- When it is offered to an agent and the agent fails to answer a call within the configured amount of time, or
- When it is offered to an agent and the call is explicitly bounced through a *Route Point Bounce Call* request.

When a call is bounced, an event is issued and the bounced call policies are applied. The bounced call policies can be configured to simply maintain the bounced call into the queue or to transfer the call to a new destination in which case it is removed from the queue. If the call is maintained in the queue, it keeps its call ID and external tracking ID.

4.3.1.2 Call Overflow Policy

A call is marked as overflowed under any of the following conditions:

- The call cannot be added to the queue because it is already full.



- The call was not handled (either by an agent or another policy) within a specific amount of time.

When a call is marked as overflowed, an event is issued and the Call Overflow policy is applied. The Call Overflow policy can be configured to perform one of the following actions:

- Provide busy treatment. If the Route Point is configured with the Call Forwarding Busy or Voice Messaging services, then the call is handled accordingly.
- Transfer the call to a new destination.
- Provide localized ringback until the call is released by the caller.

4.3.1.3 Per Call Failure Policy

A single call is marked as failed in two situations:

- When no request is received for the call within the configurable time period after being added to the queue or after the end of a treatment (including silence, Music On Hold, and ringback).
- When it is queued at a Route Point and the remote application sends a *Route Point Call Fail* request on the specific call. For more information, see section [5.4.12 Fail Call](#).

When a call is marked as failed, then one of the following two cases can occur:

- If the queued call is not already offered to an agent, then a *Route Point Call Failed* event is issued and the call is sent to the configured failed destination.
- If the failed call is offered to an agent, then it continues to be offered. If the call is eventually bounced, then a *Route Point Call Failed* event is issued and the call is sent to the configured failed destination.

4.3.1.4 Agent Whisper

The Agent Whisper is a call treatment that consists of an announcement played to the agent upon answer when a call is distributed from the Route Point. The Agent Whisper URL is optionally provided by the remote application when distributing a call to an agent.

When the agent answers the call, the whisper message is played to the agent and the following occurs:

- The *Route Point Whisper Started* event is sent.
- The caller remains connected to the Media Server, and continues hearing the media that was playing at the time of answer, if any.
- The bounce and overflow policy are no longer applied to this call.

When the whisper message finishes playing, then a *Route Point Call Answered By Agent* event is sent and the media stream is established between the caller and agent.

If the agent releases the call while hearing the whisper message, then the caller is also released. In addition, any Route Point request received while the whisper message is playing is rejected.

4.3.2 Route Point Requests

The following table provides a brief description of the requests that can be performed on a Route Point. A complete description for each request is provided in section [5.4 Route Point Requests \(AS only\)](#).

Request	Description
Distribute Call	Offer a call queued at a Route Point to an agent.
Play Treatment	Play a treatment to a selected call in the Route Point queue.
Play Music On Hold	Play Music On Hold (MOH) to a selected call in the Route Point queue.
Play ringback	Play localized ringback to a selected call in the Route Point queue.
Play silence	Play localized silence to a selected call in the Route Point queue.
Play busy	Play localized busy to a selected call in the Route Point queue.
Outgoing Dial	Originate a call session from the Route Point to the specified destination.
Bounce Call	Bounce a call queued at a Route Point to an agent.
Blind Transfer	Transfer a call queued at a Route Point to a new destination
Release Call	Release a call maintained in a Route Point queue.
Fail Call	Mark a call queued at a Route as <i>Failed</i> .
Get Route Point State	Get a Route Point state.
Modify Route Point State	Put a Route Point in the <i>Failed</i> or <i>Normal</i> state.
Get Route Point	Get the list of calls maintained in a Route Point queue.

4.3.3 Route Point Events

A series of Route Point events can be generated against a Route Point. These events are generated when a call is either added or removed to the Route Point queue. An event can also be generated to notify the completion of a request (Music On Hold has finished playing).

The following table provides a brief description of the existing Route Point events. A complete description for each event is provided in section [6.3 Conference Events](#). The last column explains how each event reflects a change in the Route Point queue content. For example, a *Route Point Call Added* event results in adding a call to the queue, while a *Route Point Call Answered by Agent* results in removing a call from the queue.

Events	Description	Queue Impact
Route Point Call Added	Event produced when a call is added to a route point.	Call added
Route Point Call Offered to Agent	Event produced when a call is offered to an agent.	No change
Route Point Call Answered by Agent	Event produced when a call offered is answered by an agent.	Call removed
Route Point Call Abandoned	Event produced when a call is released by the calling party.	Call removed
Route Point Call Overflowed	Event produced when a call cannot be added or is removed from the queue for queue overflow reasons.	Call not added (for queue size reason) or the call is removed (for time reason)

Events	Description	Queue Impact
Route Point Call Overflowed Treatment Completed	Event produced when the announcement is completed for a call treated according to the Overflow policy.	No change. Either the call was not added to the queue, or the call was already removed from the queue.
Route Point Call Transferred	Event produced when a call is transferred.	Call removed
Route Point Call Updated	Event produced when the external tracking ID or the remote party information is modified or when the Call Progress Detection (CPD) result becomes available.	No change
Route Point Call Bounced	Event produced when a call is bounced.	No change or call removed. For more information, see section 4.3.1.1 Bounced Call Policy .
Route Point Call Forwarded	Event produced when a call is forwarded because of the Forced Forwarding policy.	Call not added
Route Point Call Forwarded Treatment Completed	Event produced when the announcement is completed for a call treated according to the Forced Forwarding policy.	Call not added
Route Point Call Released	Event produced when a call is released.	Call removed
Route Point Call Failed	Event produced when a call is failed.	Call removed or call not added if it is a new incoming call.
Route Point Outgoing Call Originated	Event produced when an outgoing dial is performed.	Call added
Route Point Outgoing Call Answered	Event produced when an outgoing dial is answered by the remote destination.	No change
Route Point Treatment Started	Event produced when a treatment is started.	No change
Route Point Treatment Completed	Event produced when a treatment has ended.	No change
Route Point MOH Started	Event produced when MOH is started.	No change
Route Point MOH Completed	Event produced when MOH has ended.	No change
Route Point Ringback Started	Event produced when ringback is started.	No change
Route Point Ringback Completed	Event produced when ringback has ended.	No change
Route Point Busy Started	Event produced when busy is started.	No change
Route Point Busy Completed	Event produced when busy has ended.	No change
Route Point Silence Started	Event produced when silence is started.	No change
Route Point Silence Completed	Event produced when silence has ended.	No change

Events	Description	Queue Impact
Route Point Whisper Started	Event produced when a whisper message is started.	No change
Route Point Holiday Policy Applied	Event produced when a call is treated according to the Holiday policy.	Call not added
Route Point Holiday Policy Treatment Completed	Event produced when the announcement is completed for a call treated according to the Holiday policy.	Call not added
Route Point Night Policy Applied	Event produced when a call is treated according to the Night policy.	Call not added
Route Point Night Policy Treatment Completed	Event produced when the announcement is completed for a call treated according to the Night policy.	Call not added
Route Point Failed	Event produced when a Route Point is marked automatically or manually as failed.	Queued is emptied
Route Point Recovered	Event produced when a manual recovery is performed on a Route Point.	Queue can start to queue calls
Route Point Subscription	Event produced after a subscription is added or refreshed.	No change
Route Point Subscription Resync	Event produced whenever the Application Server shuts down abnormally allowing a remote application to resynchronize.	No change

4.3.4 DNIS

The Dialed Number Identification Service (DNIS) is the telephone number that was dialed by a caller to reach a Route Point. A Route Point can be assigned many DNIS. A DNIS name can also be associated to each DNIS number. The DNIS number and name are reported in the various Route Point events. The remote application can use this information to implement a per DNIS behavior for a specific Route Point. For instance, it can play different announcements and distribute calls to different agents based on the DNIS that was used to reach the Route Point.

4.3.5 Route Point Failure Policy

A Route Point policy is associated with a set of application controllers through configuration. For more information, see section [3.4.6 Application Controller \(AS only\)](#). A Route Point can only reach the normal of operation when at least one application controller is in the *Ready* state.

In a normal state of operation, calls received by a Route Point are queued and trigger event notification so that a remote application can take action with them. When a Route Point is taken out of this normal state of operation, the Route Point is said to be in a *Failed* state. All calls received by a “failed” Route Point are forwarded to a configured failed destination.

A Route Point can go in the *Failed* state in one of the following conditions:

- When a failure condition is detected for example, following the loss of communication with the remote application because an application controller transitions to the *Not Ready* state and no other application controllers are in the *Ready* state, see section [3.4.6 Application Controller \(AS only\)](#).
- When someone manually marks the Route Point as *Failed* through Cisco BroadWorks web portal.
- When a remote application sends a *Route Point Fail* request for the Route Point.

When a Route Point enters the *Failed* state, the following happens:

- A *Route Point Failed* event is generated.
- All calls currently queued in the Route Point and not already offered to an agent are sent to the configured failed destination in a “First In First out” algorithm. For each sent call, a *Route Point Call Failed* event is generated.
- All calls currently offered to an agent continue to be offered. If one of these calls is eventually bounced, then a *Route Point Call Failed* event is issued and the call is sent to the configured failed destination.
- All new incoming calls received are automatically routed to the failed destination. A *Route Point Call Failed* event is generated for each new call received.

A “failed” Route Point goes back to its normal state of operation when a remote application sends a *Route Point Recover* request for the Route Point.

Once a Route Point is back to its normal state of operation, all new calls received by the Route Point are queued and treated normally, all calls that were sent to the failed destination are not returned to the Route Point.

For more information on Route Point requests, see section [5.4 Route Point Requests \(AS only\)](#).

4.3.5.1 Route Point State

The following table lists the possible states for a Route Point.

State	Description
Normal	All calls received by the Route Point are queued and treated normally.
Failed	All calls received by the Route Point are sent to the failed destination.

4.3.5.2 Route Point State Table

The Route Point state behavior can be modeled using an event-driven finite state machine. This state machine is best described using the following state transition table.

The table shows how the combination of the current state (as listed in the first row) with the occurrence of a specific event (as listed in the first column) can produce a state change. For instance, when a “Route Point Failed” event occurs in the *Normal* state, the call state moves to *Failed* (as shown at the intersection of the first row and first column). A “-” means that the event cannot be generated in the specified state.

Events	Current States	
	Normal	Failed
Route Point Failed	Failed	-
Route Point Recovered	-	Normal

4.3.6 Media Playback and Digits Collection

This subsection describes how media playback and digit collection are performed.

Play-collect requests provide the ability to play prompts and detect digits for a call in a Route Point queue. The following requests are play-collect requests offered by Cisco BroadWorks:

- Route Point Play Treatment

- Route Point Play Music On Hold (MOH)
- Route Point Play Ringback
- Route Point Play Busy
- Route Point Play Silence

A play-collect request is executed in two phases: First, a prompt is played and then an (optional) digit collection is performed.

The prompt to be played is either explicitly described (as when invoking a *Route Point Play Treatment* request by specifying the audio and video URL to play) or implicitly chosen (as in the *Route Point Play Ringback* where the system ringback is played to the caller).

Digit collection is performed only if a digit map is specified in the play-collect request. The digit collection attributes (like barge-in, firstDigitTimer, interDigitTimer) are defined system wide and cannot be changed on a per request basis.

When a play-collect request is sent, a play-collect ID is returned in the response. This identifier uniquely identifies the play-collect operation initiated. The play-collect operation is then started while an event is sent to the remote application to provide the details of the operation.

A play-collect operation where no digit collection is performed can end for the following reasons:

- Because the prompt was entirely played as requested.
- Because it was implicitly or explicitly stopped. An explicit stop is produced when a new play-collect is performed. An implicit stop is produced in two cases:
 - The call was removed from the Route Point queue.
 - The call was offered to an agent and the Route Point has been configured to provide ringback to the call.
- Because an error was encountered (for example, the prompt to play was not found).

A play-collect where a digit collection is performed can end for the following reasons:

- Because the expected digit pattern was collected.
- Because the received digits did not match the digit map. This can only happen when collecting a digit sequence (that is, more than one digit). When collecting a single digit, the play-collect operation is not stopped because of an unmatched digit.
- Because no digit was collected before one of the collection timers expired
- Because it was implicitly or explicitly stopped.
- Because an error was encountered (for example, the prompt to play was not found)

When a play-collect operation is completed, an event is sent to the remote application and provides the detail of the play-collect operation.

This sequence of events is illustrated in the following time sequence diagram.

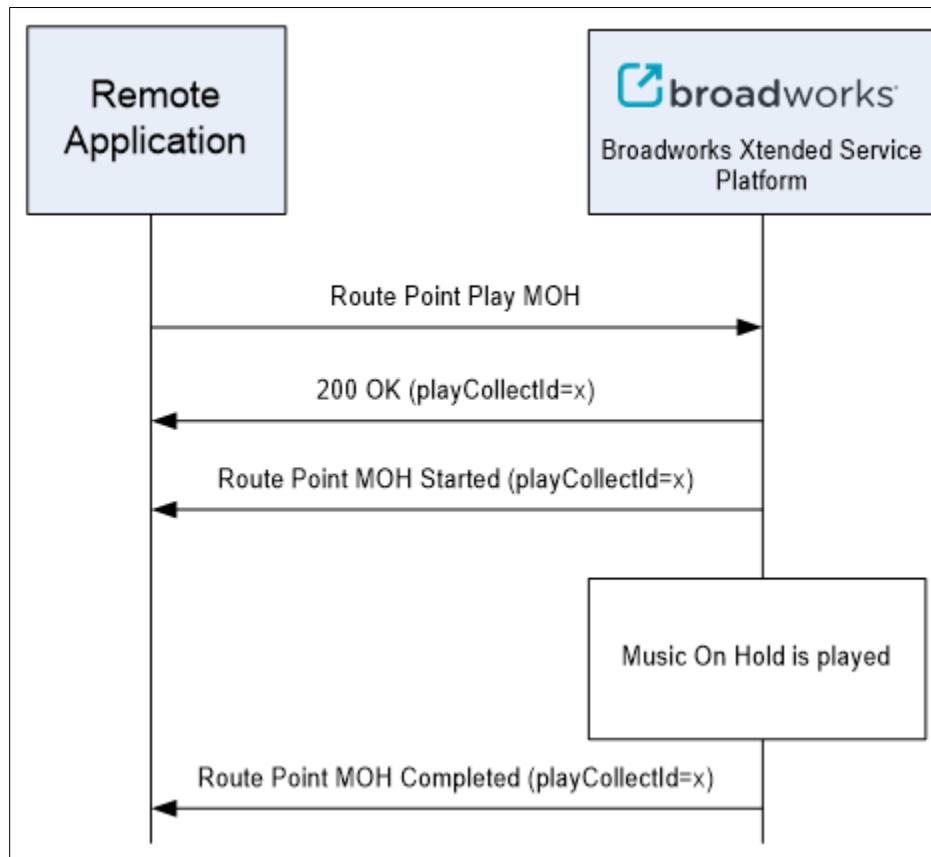


Figure 23 Media Playback and Digits Collection

NOTE: When a queue entry is being removed from the queue by policy or action, any media playback is stopped and the corresponding “completed” event is not sent. For example, if a treatment is being played while a call is offered to an agent, the treatment is stopped implicitly when the call is answered (that is, when the *RoutePointCallAnsweredByAgentEvent* is issued). However, in this case, no *Route Point Treatment Completed* event is issued.

4.4 ACD (AS only)

Automatic Call Distribution (ACD) is a service that distributes calls to agents. An ACD is identified by its subscriber ID. An ACD automatically distributes calls to agents based on their availability, while a Route Point passively maintains calls and requires a remote application to distribute them.

4.4.1 ACD Types

Cisco BroadWorks ACD provides three levels of functionality as follows:

- Basic – This option provides basic queuing capabilities distributing calls to devices and people. This is a simple option for multiple receptionists and other call center applications that require basic queuing capabilities.

- Standard – In addition to basic queuing capabilities provided by the Basic solution, this option provides such functions as agent states, supervising, and reporting.
- Premium – This option is the most feature-rich solution provided by Cisco BroadWorks. In addition to the capabilities provided by the standard offering, it provides features such as night service, stranded calls, and bounced calls routing policies.

4.4.2 Incoming Call Processing

The incoming call processing mechanism used by the ACD differs from a Route Point in the following ways:

- A series of treatments can be automatically presented to the caller when reaching the ACD. In a Route Point, it is the remote application's responsibility to explicitly play a treatment to the caller.
- Calls are ordered in the ACD queue and distributed to agents based on their priority. This order can be modified by a supervisor. In a Route Point, calls are not maintained in any specific order and it is the responsibility of the remote application to distribute the calls in the proper order.
- A caller can escape the queue by dialing the configured escape digit while a call treatment is being played. Upon dialing the escape digit, the caller is provided with Busy Treatment. If the queue is configured with the Call Forwarding Busy or Voice Messaging service, then the call is handled accordingly.
- Calls maintained in queue with no agent assigned (stranded call) can be handled appropriately. For more information, see section [4.4.2.1 ACD Bounced Call Policy](#).

The following figure illustrates how an incoming call is added to the ACD queue and when it can leave the queue.

The processing of an incoming call to an ACD can be summarized as follows:

- 1) An incoming call received by the ACD is initially processed by the services that are assigned to the ACD. In some cases, the services can prevent the call from proceeding to the queue. For example, the Call Forwarding Always service can be assigned and enabled for the ACD, in which case all incoming calls are forwarded before entering the queue.
- 2) The incoming call is then processed against the following Route Point policies:
 - Holiday Service policy
 - Night Service policy
 - Forced Forwarding policy
- 3) The incoming call enters the queue, unless the call was handled by a service or ACD policy. If the queue has reached its maximum capacity, then the call is processed according to the Overflow policy resulting in an *ACD Call Overflowed* event. Otherwise, the call is put into the queue and an *ACD Call Added* event is generated while the caller is presented with a call treatment.
- 4) Calls are distributed to agent based on the configured ACD routing policy, which can either be skill-based or priority-based routing. With skill-based routing, calls are distributed to the most skilled agent. With priority-based routing, calls are distributed to the agent based on the call priority and wait time.
- 5) A call exits the queue in one of the following conditions:
 - An agent is offered the call and answers, in which case an *ACD Call Answered by Agent* event is generated.

- The caller abandons the call by hanging up, in which case an *ACD Call Abandoned* event is generated.
 - The caller abandons the call by dialing an escape digit, in which case an *ACD Call Escaped* event is generated.
 - The supervisor transfers the call out of the call, in which case an *ACD Call Transferred* event is generated.
- A call can also exit the queue based on the processing of an ACD policy as follows:
- The Stranded Calls policy allows the configuration of specific routing behavior for stranded calls, in which case an *ACD Call Stranded* event is generated.
 - The Bounced Calls policy allows the configuration of specific routing behavior if a call offered to an available agent bounces, in which case an *ACD Call Bounced* event is generated.
 - The Overflow policy allows the configuration of specific routing behavior if the call stays in the queue beyond a configurable time threshold or if the queue has reached its maximum capacity, in which case an *ACD Call Overflowed* event is generated.

4.4.2.1 ACD Bounced Call Policy

A call is bounced under the same conditions as stated for the Route Point in section [4.3.1.1 Bounced Call Policy](#). Besides, a call can also be bounced after being put on hold by an agent for longer than a configured amount of time.

4.4.2.2 Stranded Call Policy

A stranded call is a call that is being processed by a queue that has no agents currently staffed. An agent is said to be staffing a queue if the agent has joined the queue and is not in the sign-out state. A call is stranded under any of the following conditions:

- When a call is being maintained in an ACD queue and the last agent staffing the queue "unjoins" the queue or signs out.
- When a call enters the queue and is ready to be offered to an agent (after the call has been played the entry treatment) and there are no agents staffing the queue.

When a call is stranded and the policy is enabled, an *ACD Call Stranded* event is issued, the call is then removed from the queue, and the stranded call policy is applied. The stranded call policy can be configured to perform one of the following actions:

- Provide busy Treatment. If the queue is configured with the Call Forwarding Busy or the Voice Messaging service, then the call is handled accordingly.
- Transfer the call to a new destination in which case the call is removed from the queue.
- Apply the night service policy (Premium only).
- Provide localized ringback until the call is released by the caller (Premium only).
- Play an announcement in a loop until the call is released by the caller (Premium only).

4.4.2.3 Stranded Unavailable Call Policy

Similarly to the Stranded Call Policy, the Stranded Unavailable Call Policy is applied:

- When a call is being maintained in an ACD queue and an agent state change is made such that all remaining staffed agents are now unavailable.
- When a call enters the queue and is ready to be offered to an agent (after the call has been played the entry treatment) and all staffed agents are unavailable.

When this policy is applied, an *ACD Call Stranded Unavailable* event is issued, the call is then removed from the queue, and the policy is applied. The Stranded Unavailable Call policy can be configured with the same options as the Stranded Call Policy.

4.4.2.4 Agent Whisper

The agent whisper is a call treatment that consists of an announcement that is optionally played to the agent upon answer when a call is distributed from the ACD.

When the agent answers the call, the whisper message is played to the agent and the following occurs:

- The ACD Whisper Started event is sent.
- The caller remains connected to the Media Server, and continues hearing the media that was playing at the time of answer, if any.
- The bounce and overflow policy are no longer applied to this call.

When the whisper message finishes playing, then an *ACD Call Answered By Agent* event is sent and the media stream is established between the caller and the agent.

If the agent releases the call while hearing the whisper message, then the caller is also released. In addition, any ACD request received while the whisper message is playing is rejected.

4.4.2.5 DNIS

The DNIS is the telephone number that was dialed by a caller to reach an ACD. An ACD can be assigned multiple DNIS. A DNIS name can also be associated to each DNIS number. The DNIS number and name are reported in the various ACD events.

Using DNIS numbers allows prioritizing ACD calls so that more important calls are distributed to agents first. To prioritize calls Cisco BroadWorks assigns priorities to the different DNIS numbers associated with the ACD. When a call comes in on a given DNIS number, it is assigned the priority associated with the number. In addition, announcement can be customized on a per DNIS basis.

4.4.3 ACD Requests

The following table provides a brief description of the requests that can be performed on an ACD. A complete description for each request is provided in section [5.5 ACD Requests](#).

Request	Description
Get ACD	Get the list of calls maintained in an ACD queue.
Blind Transfer	Transfer a call queued at an ACD to a new destination
Promote Call	Change the priority of a call queued in an ACD.
Reorder Call	Move the call in an ACD queue to the specified position.
Release Call	Release a call in an ACD queue.

4.4.4 ACD Events

As mentioned, a series of ACD events can be generated against an ACD. These events are generated when a call is either added or removed to the ACD queue.

The following table provides a brief description of the existing ACD events. A complete description is provided in section [6.5 ACD Events \(AS only\)](#).

The last column explains how each event reflects a change in the ACD queue content. For instance, a *ACD Call Added* event result in adding a call to the queue, while a *ACD Call Answered by Agent* result in removing a call from the queue.

Events	Description	Queue Impact
ACD Call Added	Event produced when a call is added to an ACD queue.	Call added
ACD Call Offered to Agent	Event produced when a call is offered to an agent.	No change
ACD Call Answered by Agent	Event produced when a call offered is answered by an agent.	Call removed
ACD Call Abandoned	Event produced when a call is released by the calling party.	Call removed
ACD Call Released	Event produced when a call is released.	Call removed
ACD Call Overflowed	Event produced when a call is removed from the queue for queue overflow reasons.	Call not added (for queue size reason) or the call is removed (for time reason).
ACD Call Overflowed Treatment Completed	Event produced when the announcement is completed for a call treated according to the Overflow policy.	No change. Either the call was not added to the queue, or the call was already removed from the queue.
ACD Call Transferred	Event produced when a call is transferred.	Call Removed
ACD Call Updated	Event produced when the external tracking ID or the remote party information is modified.	No change
ACD Call Bounced	Event produced when a call is bounced.	No change or call removed. For more information, see section 4.3.1.1 Bounced Call Policy .
ACD Call Forwarded	Event produced when a call is forwarded because of the Forced Forwarding policy.	Call not added
ACD Call Forwarded Treatment Completed	Event produced when the announcement is completed for a call treated according to the Forced Forwarding policy.	Call not added
ACD Call Reordered	Event produced when a call is reordered.	No change
ACD Call Stranded	Event produced when a call is stranded.	Call removed
ACD Call Stranded Treatment Completed	Event produced when the announcement is completed for a call treated according to the Stranded Call policy.	No change. The call was already removed from the queue.

Events	Description	Queue Impact
ACD Call Stranded Unavailable	Event produced when a call is stranded because all agents are unavailable.	Call removed
ACD Call Stranded Treatment Completed	Event produced when the announcement is completed for a call treated according to the Stranded Unavailable Call policy.	No change. The call was already removed from the queue.
ACD Call Escaped	Event produced when a caller has escaped by dialing the escape sequence.	Call removed
ACD Whisper Started	Event produced when a whisper message is started.	No change
ACD Holiday Policy Applied	Event produced when a call is treated according to the Holiday policy.	Call not added
ACD Holiday Policy Treatment Completed	Event produced when the announcement is completed for a call treated according to the Holiday policy.	Call not added
ACD Night Policy Applied	Event produced when a call is treated according to the Night policy.	Call not added
ACD Night Policy Treatment Completed	Event produced when the announcement is completed for a call treated according to the Night policy.	Call not added
ACD Subscription	Event produced after a subscription is added or refreshed.	No change
ACD Subscription Resync	Event produced when the Application Server shuts down abnormally allowing a remote application to resynchronize.	No change

4.4.5 Agent

An agent is a subscriber that has an agent Call Center license assigned (Basic, Standard or Premium). An agent is characterized by its subscriber ID and its state.

An agent can receive calls from an ACD or from a Route Point. The ACD “automatically” distributes calls to an agent while a Route Point distributes call to an agent when requested by a remote application.

4.4.5.1 Agent State Model

4.4.5.1.1 Agent State

The following table lists the possible agent states for agents that have the Standard or Premium license assigned.

State	Description
Sign-In	Agent is signing in. This state is a temporary state while logging in.

State	Description
Available	Agent is available.
Unavailable	Agent is unavailable.
Wrap-Up	Agent is performing post-call work.
Sign-out	Agent is signed out.

4.4.5.1.2 Agent Requests

The following table provides a brief description of the requests that can be performed on an agent. A complete description for each request is provided in section.

Request	Description
Set Agent State	Set the agent state and unavailable code.
Tag Disposition Code	Set a disposition code to last treated call.
Tag Disposition Code to callId	Set a disposition code to specified call.
Escalate to Supervisor	Originate a call session from the specified agent to an available supervisor.
Emergency Call to Supervisor	Immediately conference in a supervisor in an existing call.

4.4.5.1.3 Agent Event

A series of agent events can be generated against an agent. The following table provides a brief description of the existing call events. For a complete description, see section [6.5.25 ACD Night Policy Treatment Completed](#).

Events	Description
Agent State	Event produced when the agent state changes.
Agent Subscription	Event produced on a new subscription to provide a description of the agent.
Agent Disposition Code Added	Event produced when an agent enters a disposition code.

4.4.5.1.4 Agent State Table

A change of event state can be performed through the Set Agent State request. Any state changes are permitted. Cisco BroadWorks does not implement any server logic. For example, an agent can go from Wrap-Up to Unavailable to Sign-In. No state transition rules are enforced. The following table describes this behavior using a state transition table.

The table shows how the combination of the current state (as listed in the first row) with the occurrence of the *AgentState* event content (as listed in the first column) can produce a state change. For instance, when an *Agent State* event with the state specifying *Available* state while being in the *Sign-in* state, the agent state moves to *Available* (as shown at the intersection of the first row and second column).

AgentState Event Content	Current States				
	Sign-in	Available	Unavailable	Wrap-up	Sign-out
Agent Signed-in	Sign-in	Sign-in	Sign-in	Sign-in	Sign-in

AgentState Event Content	Current States				
	Sign-in	Available	Unavailable	Wrap-up	Sign-out
Agent Available	Available	Available	Available	Available	Available
Agent Unavailable	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable
Agent Wrapping-up	Wrap-up	Wrap-up	Wrap-up	Wrap-up	Wrap-up
Agent Signed-out	Sign-out	Sign-out	Sign-out	Sign-out	Sign-out

4.4.5.1.5 Route Point Usage of the Agent State

It is the responsibility of the remote application to track the agent state and implement any state logic when distributing a call from a Route Point. For example, Cisco BroadWorks will not prevent a Route Point from distributing a call from to an agent if their state is *Unavailable*. In this case, the remote application can decide to distribute calls to agent only when they are in the *Available* and *Wrap-up* state. In the end, the remote application can implement any business logic it finds appropriate based on the list of agent states provided.

4.4.5.1.6 ACD Usage of the Agent State

Unlike Route Point, the Cisco BroadWorks ACD tracks agent states and implements the agent state logic required by the ACD. Calls are distributed by the ACD when appropriate. However, note that the agent is allowed to transit from any state to any state. Hence, from a call distribution perspective, the unavailable state is no different from the sign-in and sign-out states as no calls are distributed from the ACD to the agent in both states.

4.4.5.1.7 Unavailable Code

The unavailable code is an additional attribute that is applied when an agent goes unavailable. It provides information regarding the agent unavailability reason (for example, agent is on a personal break, agent is on lunch, and so on.). The code can be entered on the device on the agent's desk or the agent or supervisor clients.

4.4.5.2 Disposition Code

A disposition code is an additional attribute that is applied to a call during the call and during wrap-up. It allows the call to be tagged with promotions, topics, and so on. Call Center managers can use this data to better understand the performance of their agents and call centers. The code can be added using the Cisco BroadWorks Call Center client, the Agent's endpoint or via a remote application request.

4.4.6 Supervisor

A supervisor is a Cisco BroadWorks subscriber configured to supervise agents. A supervisor has agents assigned.

4.4.7 Threshold and Alerts

Various measurements are made against Call Center queues and agents for monitoring purposes. These include for example the agent average busy time, the agent average wrap-up time, the queue estimated wait time, and so on. Threshold values can be configured against these measurements. If a measurement reaches or exceeds a configured threshold, Cisco BroadWorks reports the alert severity through events.

4.5 Call Park

The Call Park allows a subscriber to park an incoming call against their own extension or the extension of another subscriber. Subscribers within the same group or enterprise can then retrieve the call from their own phone.

A remote application can subscribe to an event package that provides information about the calls that are parked against a specified subscriber.

More information about this service is provided in the *Cisco BroadWorks Service Guide* [3].

4.6 Meet-Me Conference (AS only)

A Meet-Me conference allows a set of call sessions to be logically connected at a conference bridge. Each party connected to a Meet-Me conference is a Meet-Me conference participant. A party becomes a participant by dialing the conference bridge and entering a PIN. A Meet-Me conference is identified by the bridge subscriber ID and the conference ID.

The following figure provides a graphical representation of a Meet-Me conference with three participants: subscribers S1, S2, and S3.

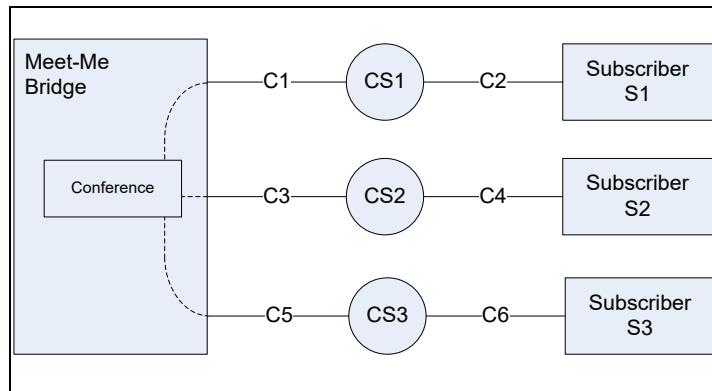


Figure 24 Meet-Me Conference

4.6.1 Meet-Me Conference Types

There are two types of Meet-Me Conferences:

- Scheduled conferences, in which a moderator schedules the conference in advance or right at the start of the conference. A unique conference ID is created for this conference and it is valid during the configured time periods. A unique moderator PIN is also generated along with the conference ID. This type of conference is most appropriate for large conferences, webinars, and recurring meetings.
- Reservationless conference, in which a moderator creates a conference without specifying a start time or an end time. A unique conference ID is assigned to this conference and it can be used to start a conference at any time. A unique moderator PIN is also generated along with the conference ID. This type of conference is most appropriate for informal meetings.

4.6.2 Roles

Host: A Cisco BroadWorks subscriber who can create, delete, and modify conferences on a conference bridge for which it is a host.

Moderator: Any subscriber who joins the conference after entering the moderator PIN is a moderator. Moderators have special privileges. They can invoke functions such as recording, locking a conference, inviting a new participant, and so on. There can be multiple moderators in a conference.

Participant: Any subscriber who joins the conference after entering the conference ID is a participant. Participants have very limited privileges and cannot control the conference. There can be multiple participants in a conference.

Attendee: An attendee may refer to either a participant or a moderator.

4.6.3 Conference Start

A conference starts in one of two ways:

- If Cisco BroadWorks is configured to wait for the moderator to start the conference, the conference starts when the moderator joins the conference. All participants who dial in before the moderator joins are placed on hold and the conference is not started. Once the moderator joins, the conference is started and all participants join the conference.
- If Cisco BroadWorks is not configured to wait for the moderator, then the conference starts when the first participant joins the conference.

4.6.4 Conference Join

In order to join a conference, a participant first dials in to the Meet-Me conference bridge. At this stage a call session is established between the caller and the Meet-Me conference bridge, however the caller is not yet connected to the conference as shown for subscriber S1 in *Figure 25*.

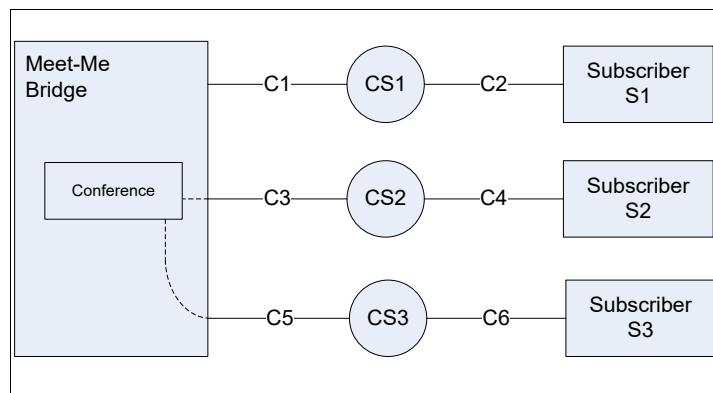


Figure 25 Subscriber S1 Connected to the Bridge but not Entered the Conference PIN

The participant is then asked to enter a PIN. If the PIN entered is the conference ID, then the caller is redirected to the conference and joins it as a simple participant. If the PIN entered is the moderator PIN, then the participant is redirected to the conference but joins as a moderator. An event is issued when a participant joins a conference.

Note that some participants can also be added through a Meet-Me conference outdial request, in which case the call session is initiated from the conference itself.

4.6.5 Capabilities

4.6.5.1 Lecture Mode

Lecture Mode is a conference policy that mutes all participants and restricts them from unmuting themselves, when it is turned on.

The Lecture mode can be enabled:

- Manually by a moderator.
- Automatically when the participant count exceeds a specific value

4.6.5.2 Recording

The Meet-Me conference audio stream can be recorded. The recording can be started, paused, resumed, and stopped either through a DTMF input or via a CTI request. Events associated to changes in recording state are issued. Each time a conference recording is stopped/started during a conference, a new recording is stored on the repository.

4.6.5.3 Lock/Unlock a Conference

An ongoing conference can be locked to prevent new attendees from joining the conference. When a conference is locked, new participants cannot join the conference. However, new attendees can be invited to join the conference using the Outdial facility.

The conference must be unlocked before new participants can join the conference. After the last moderator of a conference leaves the conference, the conference is automatically unlocked without any announcement.

4.6.5.4 Self Identify

A participant can identify themselves with a unique identifier allowing an external system to correlate the call session with its own internal information. This can be used for example by a web collaboration tool in the following way:

- 1) A conference owner creates a web collaboration session that also creates a scheduled Meet-Me conference.
- 2) A participant joins the collaboration session and is presented with the Meet-Me bridge number, the conference number, and a generated self identify number.
- 3) The user dials in to the conference bridge as normal, enters the conference ID and joins the conference.
- 4) The user enters the self identify number that is sent to the collaboration session via event notification.
- 5) The web collaboration platform can now correlate the web session with the created call session.

The self identity code can be entered via the device keypad (that is, as DTMF) during the conference or through a parameter of the Meet-Me conference outgoing dial request.

4.6.5.5 Active Talker

Cisco BroadWorks identifies the speaker of a Meet-Me conference based on loudness ranking as the active talker. There is only one active talker per conference. An event is generated when a new active talker is identified or when the current active talker stops talking.

4.6.6 Meet-Me Conferencing Request

The following table provides a brief description of the requests that can be performed on a Meet-Me Conference. For a complete description of each request, see section [5 Requests](#).

Request	Description
Get Meet-Me Conference	Get the list of participants maintained in a Meet-Me Conference.
Outgoing Dial	Originate a call session from the conference to the specified destination.
Mute Call	Mute a call participating in a conference.
Unmute Call	Unmute a call participating in a conference.
Hold Call	Hold a call participating in a conference.
Retrieve Call	Retrieve (unhold) a call participating in a conference.
Release Call	Release a call participating in a conference.
Lock	Lock the conference.
Unlock	Unlock the conference.
Start Recording	Start recording the conference.
Pause Recording	Pause the recording of a conference being recorded.
Resume Recording	Resume recording of a conference that was paused earlier.
Stop Recording	Stop recording the conference.
Set Lecture Mode	Set the conference to lecture mode.
Unset Lecture Mode	Unset the lecture mode
Play Security Classification	Play a message that announces the current security classification level calculated for the conference.
Set Conference Client Session Info	Associate the client session info with the existing conference.

4.6.7 Meet-Me Conferencing Events

A series of conference events can be generated against a Meet-Me conference. The following table provides a brief description of the existing conference events. For a complete description of each event, see section [6 Events](#).

Events	Description
Meet-Me Conference Started	Event produced when a Meet-Me conference is started.
Meet-Me Conference Stopped	Event produced when a Meet-Me conference is stopped.
Meet-Me Conference Locked	Event produced when a Meet-Me conference is locked.
Meet-Me Conference Unlocked	Event produced when a Meet-Me conference is unlocked.
Meet-Me Conference Recording Started	Event produced when a Meet-Me conference recording is started.
Meet-Me Conference Recording Stopped	Event produced when a Meet-Me conference recording is stopped.

Events	Description
Meet-Me Conference Recording Paused	Event produced when a Meet-Me conference recording is paused.
Meet-Me Conference Recording Resumed	Event produced when a Meet-Me conference recording is resumed.
Meet-Me Conference Lecture Mode Started	Event produced when a Meet-Me conference lecture mode is started.
Meet-Me Conference Lecture Mode Stopped	Event produced when a Meet-Me conference lecture mode is stopped.
Meet-Me Conference Auto Lecture Mode Started	Event produced when a Meet-Me conference auto lecture mode is started.
Meet-Me Participant Joined	Event produced when one or more participants join a Meet-Me conference.
Meet-Me Participant Left	Event produced when one or more participants leave a Meet-Me conference.
Meet-Me Participant Muted	Event produced when one or more participants are muted in a Meet-Me conference.
Meet-Me Participant Unmuted	Event produced when one or more participants are unmuted in a Meet-Me conference.
Meet-Me Participant Held	Event produced when one or more participants are being held in a Meet-Me conference.
Meet-Me Participant Retrieved	Event produced when one or more participants are being unheld in a Meet-Me conference.
Meet-Me Participant Updated	Event produced when one or more participants have their participant information updated in a Meet-Me conference.
Meet-Me Subscription	Event produced after a subscription is added or refreshed.
Meet-Me Subscription Resync	Event produced when the Application Server shuts down abnormally allowing a remote application to resynchronize.
Meet-Me Conference Security Classification Updated	Event produced when there is a change to the security classification level of a Meet-Me conference.
Meet-Me Conference Client Session Info Updated	Event produced when there are changes to the client session info for the conference.

4.7 Collaborate Room (AS only)

A Collaborate Room allows a set of call sessions to be logically connected at a conference bridge, optionally with video. Each party connected to a Collaborate Room is a Collaborate Room participant. A party becomes a participant when invited to join by the outdial capability or by dialing the Collaborate Room phone number and entering the room ID.

The following figure provides a graphical representation of a Collaborate Room with three participants: subscribers S1, S2, and S3.

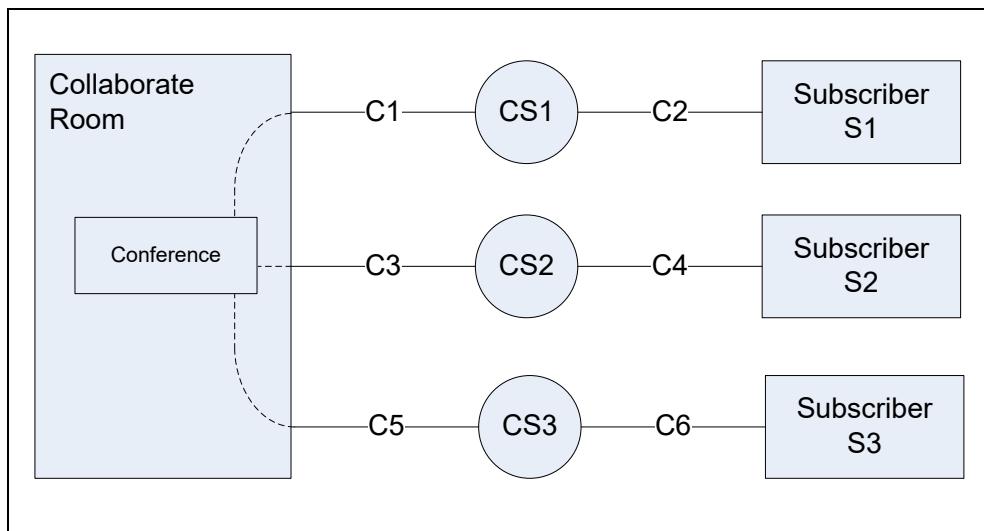


Figure 26 Collaborate Room

4.7.1 Collaborate Room Types

There are three types of Collaborate Rooms:

- My Room: This is a default multi-user chat room available to all users that have the Collaborate – Audio service assigned. This room is available to all users and is advertised via the enterprise directory.
- Project Rooms: These are rooms created for a specific topic. These rooms can have indefinite life spans like *My Room* or support one-time or schedulable sessions.
- Instant Rooms: These rooms are created as needed by the client. They only exist for as long as desired by the user that created the room.

4.7.2 Roles

There are two types of roles:

- Owner: The subscriber that owns the room. The owner invites other subscribers to join the room and can also invoke functions such as locking a room or muting a participant. There is only one owner in a room.
- Participant: Any subscriber who joins the room either by being invited or by dialing to the room and entering the room ID. There can be multiple participants in a room.

4.7.3 Collaborate Room Start

A room starts in one of two ways:

- If Cisco BroadWorks is configured to wait for the owner to start the room, the room starts when the owner joins the room. All participants who dial in before the owner joins are placed on hold and the room is not started. Once the owner joins, the room is started and all participants join the room.
- If Cisco BroadWorks is not configured to wait for the owner, then the room starts when the first participant joins the room.

4.7.4 Collaborate Room Join

In order to join a room, a participant first dials in to the Collaborate Room. At this stage, a call session is established between the caller and the Collaborate Room; however, the caller is not yet connected to the conference as shown for subscriber S1 in *Figure 25*.

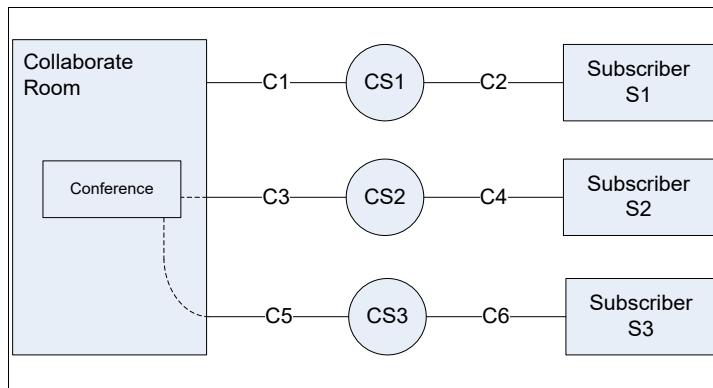


Figure 27 Subscriber S1 Dialed into the Collaborate Room but not Entered the Room ID.

The participant is then asked to enter a room ID. If the ID entered is the room ID, then the caller is redirected to the specific room and joins it as a simple participant. An event is generated when a participant joins a conference.

Note that some participants can also be added through a Collaborate Room outdial request, in which case, the call session is initiated from the conference itself.

4.7.5 Capabilities

4.7.5.1 Lock/Unlock a Room

An active room can be locked to prevent new attendees from joining the room. When a room is locked, new participants cannot join the room by dialing in or using the outdial capability.

The conference must be unlocked before new participants can join the room.

4.7.5.2 Muting/Unmuting a Participant

A participant can be muted or unmuted by the owner.

4.7.5.3 Active Talker

Cisco BroadWorks identifies the speaker of a Collaborate Room based on loudness ranking as the active talker. There is only one active talker per room. An event is generated when a new active talker is identified or when the current active talker stops talking.

4.7.6 Collaborate Room Request

The following table provides a brief description of the requests that can be performed on a Collaborate Room. For a complete description of each request, see section [5 Requests](#).

Request	Description
Get Collaborate Room	Get the list of participants present in a Collaborate Room.
Outgoing Dial	Originate a call session from the room to the specified destination.
Mute Call	Mute a call participating in a room.

Request	Description
Unmute Call	Unmute a call participating in a room.
Release Call	Release a call participating in a room.
Lock	Lock the active room.
Unlock	Unlock the active room.
Set Conference Client Session Info	Associate the client session info with the active room.

4.7.7 Collaborate Room Events

A series of room events can be generated against a Collaborate Room. The following table provides a brief description of the existing room events. For a complete description of each event, see section [6 Events](#).

Events	Description
Collaborate Room Started	Event produced when a Collaborate Room is started.
Collaborate Room Stopped	Event produced when a Collaborate Room is stopped.
Collaborate Room Locked	Event produced when a Collaborate Room is locked.
Collaborate Room Unlocked	Event produced when a Collaborate Room is unlocked.
Collaborate Room Participant Joined	Event produced when one or more participants join a Collaborate Room.
Collaborate Room Participant Left	Event produced when one or more participants leave a Collaborate Room.
Collaborate Room Participant Muted	Event produced when one or more participants are muted in a Collaborate Room.
Collaborate Room Participant Unmuted	Event produced when one or more participants are unmuted in a Collaborate Room.
Collaborate Room Updated	Event produced when one or more participants have their participant information updated in a Collaborate Room.
Collaborate Room Subscription	Event produced after a subscription is added or refreshed.
Collaborate Room Subscription Resync	Event produced when the Application Server shuts down abnormally allowing a remote application to resynchronize.
Collaborate Room Client Session Info Updated	Event produced when there are changes to the client session info for the Collaborate Room.



4.8 Service Management

A remote application can retrieve and set some of the subscriber service configuration in Cisco BroadWorks. It can also receive event notification following service configuration changes.

The following subsections provide more details on the various services available through the CTI interface. It also lists the requests used to modify these services along with their associated events.

4.8.1 Service Description

4.8.1.1 Call Forwarding Always

The Call Forwarding Always (CFA) enables the redirection of all incoming calls unconditionally (that is, busy, idle, alerting, and so on) to another phone number. Furthermore, a ring splash (500 ms ring burst) can be applied to the subscriber's device each time a call is forwarded.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.2 Call Forwarding No Answer

This Call Forwarding No Answer (CFNA) enables a subscriber to redirect all unanswered incoming calls to another phone number. A call is considered unanswered if it is in the *Alerting* state for a configured duration (specified in number of rings).

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.3 Call Forwarding Busy

The Call Forwarding Busy (CFB) enables a subscriber to redirect incoming calls to another phone number when the subscriber is busy. A subscriber is considered busy when there are too many active calls or a service makes the subscriber appears busy to the caller (for example, services such as Do Not Disturb or Selective Call Rejection).

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.4 Do Not Disturb

The Do Not Disturb (DND) service allows subscriber to set their status as unavailable. When a subscriber activates the Do Not Disturb (DND) service, all calls to the subscriber are processed as if the subscriber is busy and cannot receive calls. Other terminating services trigger on the busy condition, as if the subscriber is really busy.

Since the usual busy processing applies to the call, the caller is unaware that the subscriber has the service activated. Every time a call is blocked or deflected as a result of the service, the subscriber can be played a ring splash as a reminder that the service is active.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.5 Call Transfer

The Call Transfer service provides the ability to transfer a call using one of the following means:

- flash-based services
- CommPilot Call Manager
- subscriber endpoint
- CTI interface



The Call Transfer service can be configured for Call Transfer Recall. When Call Transfer Recall is configured, you can ensure that the transferred party is handled by a live person, either the intended destination or the receptionist. Call Transfer Recall also prevents diversion by the transfer destination, to avoid sending the calling party to voice mail and to ensure that Call Transfer Recall is activated if the transfer destination is unavailable.

The Call Transfer service also has an option that allows a transferring user to camp on a call against a busy destination. If the destination party becomes idle within a configurable time frame, the camp-on call alerts the transfer-to party. If the camp-on call remains unanswered beyond the configurable time frame, the camp-on call recalls the transferring party.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.6 Call Waiting

Call Waiting allows you a subscriber to receive another call while being on the phone. When an incoming call is received while a user is already engaged in a call, the subscriber is informed of the new call via a call waiting tone. The subscriber may answer the incoming call and hold the original party. The subscriber may subsequently toggle between the two parties.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.7 Last Number Redial

Last Number Redial allows a subscriber to repeat the last call that was made.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.8 Music On Hold

Music On Hold allows a subscriber to play music to callers when the call is held or parked.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.9 N-Way Call

N-Way Call allows a subscriber to create a conference, up to a maximum number configured at the system or service provider/enterprise level. When the user who created a conference releases the call, all participants, (as well as the conference) are released.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.10 Third-Party Voice Mail Support

Third-Party Voice Mail Support facilitates the support and integration of an external voice mail platform. This capability is required to deploy Cisco BroadWorks with a third-party voice mail platform, while retaining the integration of voice mail with other Cisco BroadWorks services. It allows forwarding of busy and/or unanswered calls to an external voice messaging platform.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.11 Three-Way Call

Three-Way Call allows a subscriber to create a conference with two other parties. When the user who created a conference releases the call, all participants, (as well as the conference) are released.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].



4.8.1.12 Voice-Messaging

Voice-Messaging enables a subscriber to record messages from callers for calls that are not answered. It also allows:

- Message delivery to any specified e-mail account
- Message waiting notification delivered to the phone and to any specified mail or short message service (SMS) account (for example, cell phone)
- Integration of the messaging capabilities with Cisco BroadWorks services (Call Back, Transfer, CommPilot Express, escape to extension, voice portal, and so on)
- Integration of hybrid messaging systems within an enterprise
- Administrator and user self-management through the web portal

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.13 Sequential Ring

Sequential Ring allows a subscriber to define a “find-me” list of phone numbers, which are alerted sequentially upon receiving an incoming call that matches a set of criteria. While the service searches for the subscriber, the calling party is provided with a greeting followed by periodic comfort announcements. The caller can also interrupt the search at any point to leave a message by pressing a DTMF key.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.14 Simultaneous Ringing

This service enables a user to have multiple destinations ring simultaneously when any calls are received on their phone number. The first destination to be answered is connected.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.15 BroadWorks Anywhere

BroadWorks Anywhere provides the ability to define one or more network locations (also referred as BroadWorks Anywhere locations) that can be used as extensions to the subscriber's account. BroadWorks Anywhere allows subscriber to make and receive calls from any device, at any location, with only one phone number, one dial plan, one voice mailbox, and a unified set of features. It can also be used to move a call from one phone to another phone.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.16 Voice Mail Message Summary

Cisco BroadWorks provides the ability to get and be notified of message summary information changes for a subscriber voice mail. Summary information provides information on the number of old and new messages and if any message has been flagged as urgent. A remote application can also clear the message summary content.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].



4.8.1.17 Hoteling (**AS only**)

Hoteling allows subscribers to associate their service profiles with a different device. It is typically used for transient employees, that is, an enterprise can set up a visitor cube with a phone, and a visiting employee can use the phone with the visiting employee's service profile. Hoteling functionality is delivered via two separate services, Hoteling Host and Hoteling Guest.

- Subscriber accounts with Hoteling Host assigned allow their devices to host other Hoteling Guest subscribers.
- Subscribers that have Hoteling Guest assigned can be associated with a Hoteling Host subscriber and use the Hoteling Host subscriber's device with their service profiles.

Typically, a guest gets the list of available hoteling host stations it can use. Using this list, the subscriber selects one of the available hosts and associates his service profile to it. An event is produced on a host/guest association change to notify a remote application of the result.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.18 Call Recording (**AS only**)

Call Recording provides the ability to record calls. The service can be configured to never record calls, always record calls, or initiate call recording on demand following a subscriber request. When a call is recorded, the recording file is saved on a recording platform and can be retrieved by the subscriber at a later time.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.19 Remote Office (**AS only**)

Remote Office allows a subscriber to access and use their Cisco BroadWorks profile and services from any device. It enables subscribers to use all of their features while working remotely (for example, extension dialing, transfers, conference calls, Outlook Integration, directories, and so on). In addition, since calls are still originated from Cisco BroadWorks, the service provides an easy mechanism for separating personal and business phone expenses, as well as keeping alternate phone numbers private.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.20 CommPilot Express (**AS only**)

CommPilot Express allows a subscriber to pre-configure multiple profiles for managing incoming calls differently. The four profiles are:

- Available – In the office
- Available – Out of the office
- Busy
- Unavailable

Each profile includes preferences for managing the relevant incoming call functions (for example, Call Forwarding [busy, no answer, always, selective], Voice Messaging, Simultaneous Ringing, Call Notify), which can be configured through a single web page.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.1.21 Flexible Seating

A Flexible Seating Host is a virtual subscriber provisioned with a phone device. A user with the Flexible Seating Guest service enabled can create an association with the host in the same group (or enterprise depending on the access level of the Flexible Seating Host). After the host-guest association is established, the host phone device is reconfigured with the guest user's device settings and becomes guest's alternate device.

For more information regarding this service, see the *Cisco BroadWorks Service Guide* [3].

4.8.2 Service Management Requests

The following table provides a brief description of the requests that can be performed on the services. A complete description for each request is provided in section [5.9 Service Management Requests](#).

Request	Description
Modify Call Forwarding Always	Set or modify the Call Forwarding Always service configuration.
Get Call Forwarding Always	Get the Call Forwarding Always service configuration.
Modify Call Forwarding No Answer	Set or modify the Call Forwarding No Answer service configuration.
Get Call Forwarding No Answer	Get the Call Forwarding No Answer service configuration.
Modify Call Forwarding Busy	Set or modify the Call Forwarding Busy service configuration.
Get Call Forwarding Busy	Get the Call Forwarding Busy service configuration.
Modify Do Not Disturb	Modify the Do Not Disturb service configuration.
Get Do Not Disturb	Get the Do Not Disturb service configuration.
Clear Voice Mail Message Summary	Clear the voice mail message summary of the Third-Party Voice Mail Support, Third-Party MWI Control, or Voice Messaging service.
Get Voice Mail Message Summary	Get the voice mail message summary of the Third-Party Voice Mail Support, Third-Party MWI Control, or Voice Messaging service.
Modify Hoteling Guest (<i>AS only</i>)	Modify the Hoteling Guest service configuration.
Get Hoteling Host List (<i>AS only</i>)	Get the list of available Hoteling Host for a Guest.
Get Hoteling Guest (<i>AS only</i>)	Get the Hoteling Guest service configuration.

4.8.3 Service Management Events

As mentioned, a series of service management events can be generated against a subscriber. These events are typically generated when the service configuration is modified or after a subscription is added or refreshed.

A complete description is provided in section [6.13 Service Management Events](#).

Events	Description
Call Forwarding Always	Event produced when the Call Forwarding Always service configuration is modified.
Call Forwarding No Answer	Event produced when the Call Forwarding No Answer service configuration is modified.



Events	Description
Call Forwarding Busy	Event produced when the Call Forwarding Busy service configuration is modified.
Do Not Disturb	Event produced when Do Not Disturb service configuration is modified.
Call Transfer	Event produced when the Call Transfer service configuration is modified.
Call Waiting	Event produced when the Call Waiting service configuration is modified.
Music On Hold	Event produced when the Music On Hold service configuration is modified.
N-Way Calling	Event produced when the N-Way Calling service configuration is modified.
Third-Party Voice Mail Support	Event produced when the Third-Party Voice Mail Support service configuration is modified.
Three-Way Calling	Event produced when the Three-Way Calling service configuration is modified.
Voice-Messaging	Event produced when the Voice Messaging service configuration is modified.
Sequential Ringing	Event produced when the Sequential Ringing service configuration is modified.
Simultaneous Ringing Personal	Event produced when the Simultaneous Ringing Personal configuration is modified.
Voice Mail Message Summary	Event produced when the voice mail content is modified.
Hoteling Guest (AS only)	Event produced when the Hoteling Guest service configuration is modified.
BroadWorks Anywhere (AS only)	Event produced when the BroadWorks Anywhere service configuration is modified.
Remote Office (AS only)	Event produced when the Remote Office service configuration is modified.
CommPilot Express (AS only)	Event produced when the CommPilot Express service configuration is modified.
Flexible Seating (AS only)	Event produced when the Flexible Seating Guest configuration is modified.

4.9 Voice Messaging (AS only)

Cisco BroadWorks provides the ability to retrieve, modify, and delete voice messages.

Calls to the voice messaging commands may be rejected by Cisco BroadWorks if the frequency of requests exceeds the permitted thresholds either for a single user or for all users.

4.9.1 Voice Messaging Request

The following table provides a brief description of the Voice Messaging requests. These commands only apply to users with the Voice Messaging service. It does not apply to the Third-Party Voice Mail Support service. For a complete description of each request, see section [5 Requests](#).

Request	Description
Delete Voice Messaging Message	Delete a voice message.
Get Voice Messaging Messages	Retrieve the full content of a voice message.
Get All Voice Messaging Messages	Retrieve a list of detailed information regarding a subscriber's voice messages.
Mark All Voice Messaging Messages As Unread	Mark all user messages as "new". That is, not "read" yet.
Mark All Voice Messaging Messages As Read	Mark all user messages as "read". That is, not "new".
Mark Voice Messaging Message As Unread	Mark a message as "new". That is, not "read" yet.
Mark Voice Messaging Message As Read	Mark a message as "read". That is, not "new".

5 Requests

Requests are commands used to invoke functionality from Cisco BroadWorks. Cisco BroadWorks returns a response to every request received. Every request/response contains common parameters. The details of the request and response format are described in the following tables.

5.1 Requests and Responses

5.1.1 Request

While there are several request types that can be sent by a remote application, all requests share a common set of parameters. The following table lists these common parameters.

Parameter	Parameter Type	Required	Description
<i>requestId</i>	RequestId	Yes	An identification supplied by the remote application to map request to response.
<i>credential</i>	Base64Binary	Yes	Authenticated subscriber credentials.
<i>sessionId</i>	SessionId	No	The session identifier. This value was returned by Cisco BroadWorks in a previous response when the session was established. It is recommended to include this parameter whenever the request uses the same authentication credentials to eliminate the authentication overhead of a processing request.
<i>version</i>	String	No	The software version associated with this request.

Each specific request type contains these common parameters followed by some request specific parameters.

5.1.2 Response

Cisco BroadWorks uses the Response type to return the details of a response to a request. The Response type always contains the following common set of parameters.

Parameter	Parameter Type	Required	Description
<i>requestId</i>	RequestId	Yes	An identification supplied by the remote application to map the request to the response.
<i>sessionID</i>	SessionId	No	The session identifier.
<i>statusCode</i>	Integer	Yes	The status code indicates the result of the request, and is used by the remote application to determine whether the requested action has succeeded or not. Status codes correspond to the values defined in the <i>RFC 2616 Hypertext Transfer Protocol-HTTP/1.1</i> [11].
<i>reason</i>	String	Yes	The reason is included for informational purposes and corresponds to the status codes defined in <i>RFC 2616 Hypertext Transfer Protocol-HTTP/1.1</i> [11].
<i>errorInfo</i>	ErrorInfo	No	Present in case of a failure response. This field can be used to obtain additional information on the source of the failure.



The Response type can also contain some additional parameters specific to a request response. More information is provided in the following subsections.

The *errorInfo* is present in case of a failure and contains the following parameters.

Parameter	Parameter Type	Required	Description
<i>errorCode</i>	Integer	Yes	The code of the error.
<i>summary</i>	String	Yes	The description of the error. Contains the error text in the authenticated subscriber's language.
<i>summaryEnglish</i>	String	Yes	The description of the error in English.

In case of a failure, the *errorInfo* is not always present in the returned error message. The following table lists the various status codes that are returned by this application and whether the *errorInfo* is present or not for each case.

statusCode	Reason Value	errorInfo	Scenario(s)
400	Bad request	Mandatory	Cannot understand request, bad syntax. Generally, most CTI errors fall under this category.
401	Unauthorized	Optional	Incorrect user/user out of scope in Xsi-URL.
403	Forbidden	Optional	Access is forbidden according to the configuration.
404	Not found	Optional	Unknown URL. (This is the wrong version, service name, or resource name in the Xsi-URL.)
405	Method not allowed	Mandatory	Unsupported method is received.
409	Conflict	Mandatory	Could not fulfill request due to resource conflict. (Invalid, not well formed, or missing XML payload.)
412	Precondition failed	Mandatory	Precondition given by client failed.
500	Internal server error	Optional	Unexpected server errors.
503	Service unavailable	Optional	Requested resource is not available currently.

5.1.3 Common Errors

Since some errors are common to multiple requests, they have been regrouped in this section and are referred by each request in the following subsections. A complete list of all error code is provided in section [10 Error Code List](#).

5.1.3.1 Target User Errors

statusCode	errorCode	Summary
400	100002	Invalid user ID <userId>. The user ID must be of the form user@domain.
400	100003	User <userId> not found.
400	100014	Admin/User <userId> is not authorized for user <userId>.
400	100011	Blocked by the user's Privacy service.



statusCode	errorCode	Summary
400	100015	User is a virtual subscriber.

5.1.3.2 Address Error

statusCode	errorCode	Summary
400	100005	Address <address> must be a SIP or tel URI.

5.1.3.3 CallId Errors

statusCode	errorCode	Summary
400	101006	Call <callId> is not valid.
400	101002	Call <callId> not found.

5.1.3.4 Address Destination Errors

statusCode	errorCode	Summary
400	100001	User with address <address> is not found.
400	100012	User for address <address> has no primary phone number.
400	100013	Address <address> cannot be an international E.164 address.

5.1.3.5 System Errors

The system errors are usually caused by invalid configuration, system failure, and so on. These errors are divided in three groups.

In the first group, errors do not have summary or an errorCode associated with them. They are generated by low-level software components.

statusCode	Reason
401	Not authorized
403	Forbidden
404	Not found
500	Internal server error
503	Service unavailable

The second group is composed of system errors that have some summary information and an errorCode associated. However, the error summary for this group however cannot be localized.

statusCode	errorCode	Summary
400	1000	Missing required parameter.
500	1001	Invalid response from server.
400	1100	Invalid value of <i>startIndex</i> or number of results. Minimum value of <i>startIndex</i> and number of results should be 1.

statusCode	errorCode	Summary
503	1101	Non-HTTPs request.
409	1102	Missing entity body in request.
409	1103	Failed to parse XPath information.
400	1104	Unexpected value for format.
400	1105	Callback not acceptable.
400	1106	Callback expected.
500	1107	Internal server error, session ended before request processing.
405	1108	Method not supported.
409	1109	Unable to read from request.
404	1110	User name missing in URL.
500	1111	Internal server error – unexpected error while reading from request.
409	1112	Parsing xml-fragment body fails.
404	1113	Service handler not found.
400	1114	Unable to read JSON from request.
500	1115	Unexpected exception while providing JSON response.
409	1116	Invalid Xml.

The third and final group is composed of system errors that have summary information and an errorCode associated with them. The summary information can be localized for these errors.

statusCode	errorCode	Summary
500	102000	System error.
503	102001	Denied due to overload.
400	102002	Unable to find or create session.
400	102003	XML parsing failed: <message>.
400	102004	Invalid encoding: <encoding>.
400	102005	XML schema validation failed: <errorList>.
400	102006	Request is not valid.
400	102007	Request handler not found.
400 or 403 ¹	100004	Unauthorized request.

5.1.3.6 Subscription Errors

statusCode	errorCode	Summary
403	110040	The user has not been assigned the BroadWorks Anywhere service.
403	110200	The user has not been assigned the Call Center-Basic, Call Center-Standard, or Call Center-Premium service.

¹ Error 100004 "Unauthorized request" is associated with a status code of 403 when returned by subscription, channel, and application controller requests. All other requests use a status code of 400 for this error.

statusCode	errorCode	Summary
403	110202	The user has not been assigned the Call Center-Premium service.
403	110203	The user has not been assigned to a Call Center.
403	110070	The user has not been assigned the Call Forwarding Always service.
403	110080	The user has not been assigned the Call Forwarding Busy service.
403	110090	The user has not been assigned the Call Forwarding No Answer service.
403	110500	The user has not been assigned the Call Transfer service.
403	110570	The user has not been assigned the Call Waiting service.
403	110630	The user has not been assigned the CommPilot Express service.
403	110690	The user has not been assigned the Do Not Disturb service.
403	110710	The user has not been assigned the Hoteling Guest service.
403	110720	The user has not been assigned the Last Number Redial service.
403	110730	The user has not been assigned the Music On Hold User service.
403	110740	The user has not been assigned the N-Way Calling service.
403	110820	The user has not been assigned the Remote Office service.
403	110407	The user has not been assigned a Route Point.
403	110750	The user has not been assigned the Sequential Ringing service.
403	110770	The user has not been assigned the Simultaneous Ringing Personal service.
403	110780	The user has not been assigned the Third-Party Voice Mail Support service.
403	110790	The user has not been assigned the Three-Way Calling service.
403	110801	The user has not been assigned the Voice Messaging user service.
403	110980	The user has not been assigned the Call Recording service.

5.2 Call Management Requests

5.2.1 Dial

5.2.1.1 Description

This request is used to originate a call session from the specified subscriber to the specified destination. Cisco BroadWorks performs basic validation of the request and, if it is accepted, then it assigns a call ID and an external tracking ID to the connection and returns it in the response.

In the context of processing the dial request, Cisco BroadWorks initiates a call with the endpoint of the subscriber that initiated the dial. Cisco BroadWorks alerts the subscriber's endpoint prompting the subscriber to go off-hook.

Depending on the endpoint capabilities, Cisco BroadWorks can force an endpoint to go off-hook (that is, via the speakerphone) in two cases:

- When receiving a Talk request from a remote application. This is possible if the endpoint supports the remote control talk event package. For more information, see the *Cisco BroadWorks SIP Access Side Extensions Interface* [8].
- When configured to perform auto-answer. This is possible if the endpoint supports the SIP answer-after parameter of the SIP Call-Info header. For more information, see the *Cisco BroadWorks SIP Access Side Extensions Interface* [8].

Once the subscriber endpoint has gone off-hook, either manually or through one of the forced answer methods mentioned above, the call to the specified destination is resumed by Cisco BroadWorks.

Note that a dial request can be performed if the subscriber is already involved into a call, in which case the existing call is placed on hold.

The following diagrams provide information on the state and topology changes involved.

5.2.1.2 Case 1: Simple Dial

Pre-conditions: Subscriber S1 and subscriber S2 are in the *Idle* state.

Request: The remote application sends a Dial request for subscriber S1 with the destination phone number of S2. When BroadWorks receives the request, it first validates it and, if it is valid, creates the call C1 and Call Session CS1 and return the callId and externalTrackingId to the remote application. Subscriber S1 goes off-hook and a call C2 is established to subscriber S2.

Post-conditions: Call C1 and C2 are in the Alerting state.

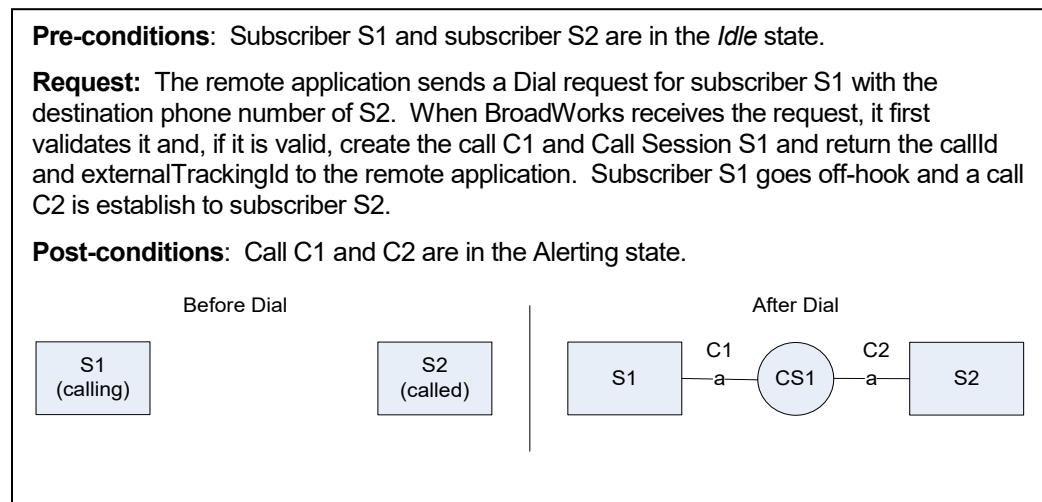


Figure 28 Simple Dial

5.2.1.3 Case 2: Dial with Existing Active Call

Pre-conditions: Subscriber S1 is connected to Call Session CS1 through call C1. Call C1 is in the Active state.

Request: The remote application sends a Dial request for subscriber S1 with the destination phone number of S3. When BroadWorks receives the request, it first validates it and creates the call C3 and Call Session CS2 and returns the callId and externalTrackingId to the remote application. Subscriber S1 accepts the call and a new call C4 to subscriber S3 is created.

Post-conditions: The call C3 and C4 are in the Alerting state. Call C1 and C2 are in the Held and Remote Held state.

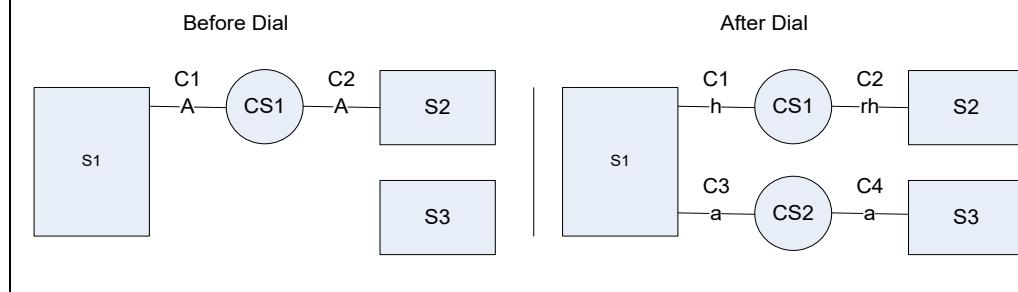


Figure 29 Dial with Existing Active Call

5.2.1.4 Request

A remote application can issue this request by sending a Dial Request. A Dial Request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the calling user.
<i>address</i>	Address	Yes	Address to call.
<i>location (AS only)</i>	ClickToDialLocationEnum	No	When present, specifies the location of the calling user to ring for the call. When not present, locations are alerted per the calling user's profile on Cisco BroadWorks.



Parameter	Parameter Type	Required	Description
<i>locationAddress</i> (AS only)	NonEmptyToken	No	<p>Specific address of location to ring. This element should only be present if the "location" element is present and set to "BroadWorksAnywhere", "SharedCallAppearance", or "Mobility".</p> <p>When the "location" element is set to "BroadWorksAnywhere", the locationAddress is optional and contains a directory number. The directory number may be in E.164 format or it may just contain digits. When not present, all BroadWorks Anywhere locations are alerted. The following are valid directory numbers for the locationAddress element:</p> <ul style="list-style-type: none"> ▪ 9725551000 ▪ +19725551000 <p>When the "location" element is set to "SharedCallAppearance", the locationAddress is optional and contains a Cisco BroadWorks address of record. The address of record can be user@domain for SIP locations or user for Media Gateway Control Protocol (MGCP) locations. When not present, all Shared Call Appearance locations are alerted. The following are valid address of records for the locationAddress element:</p> <ul style="list-style-type: none"> ▪ 9725551000@broadsoft.com ▪ 9725551000 <p>When the "location" element is set to "Mobility", the locationAddress is optional and contains a directory number. The directory number may be in E.164 format or it may just contain digits. When not present, all BroadWorks Mobility locations are alerted. The following are valid directory numbers for the locationAddress element:</p> <ul style="list-style-type: none"> ▪ 9725551000 ▪ +19725551000
<i>info</i> (AS only)	String	No	This is an opaque information provided by the client for the Application Server to correlate this information with the call session. The Application Server stores this information against the call session and is transmitted to both ends of the call session during call setup.
<i>conferenceId</i> (AS only)	String	No	This is the conference ID for the conference to join.
<i>pin</i> (AS only)	String	No	This is the security PIN for the conference to join, if the conference requires a security PIN.

5.2.1.5 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>callStartInfo</i>	CallStartInfo	No	Contains the call ID and external Tracking ID of the new call. It is only present when the statusCode is set to "201".

5.2.1.6 Associated Events

5.2.1.6.1 Case 1

The Case 1 is broken down in three sub-cases depending on how the call is answered as follows:

- Call answered by manually going off-hook
- Call answered through a Talk request
- Call automatically answered

Call answered by manually going off-hook

The following diagram illustrates the request and events exchanged when Subscriber S1 endpoint is manually going off-hook.

When a dial request is accepted, a *Call Originating* event is generated indicating that the subscriber's endpoint is being alerted. When the calling subscriber goes off-hook, first a *Hook Status* event is issued followed by a *Call Originated* event indicating that the call session origination is proceeding to the specified destination.

A *Call Updated* event is generated after the *Call Originating* event if the subscriber's endpoint supports the protocol extensions to immediately answer without requiring a subscriber action. In such a case, the *allowAnswer* parameters will be set in the event. If the endpoint does not support this extension then no event is generated.

If the calling user is not available (for example, busy, device not registered, and so on.) then a *Call Released* event is sent after the *Call Originating* event with the appropriate *releaseCause*.

If the call cannot be established (for example, the called number is invalid), then a *Call Originating* event is first sent followed by a *Call Originated* event. A *Call Releasing* event is generated with the *releaseCause* set as the subscriber is connected to a media treatment. A *Call Released* event is generated once the treatment stops playing or if the subscriber hangs up. The event also contains the same *releaseCause* value.

Call answered through a Talk request

If the subscriber's endpoint supports the remote control talk event package, then a remote application can force an endpoint off-hook, without a subscriber intervention. The following diagram illustrates the request and events exchanged.

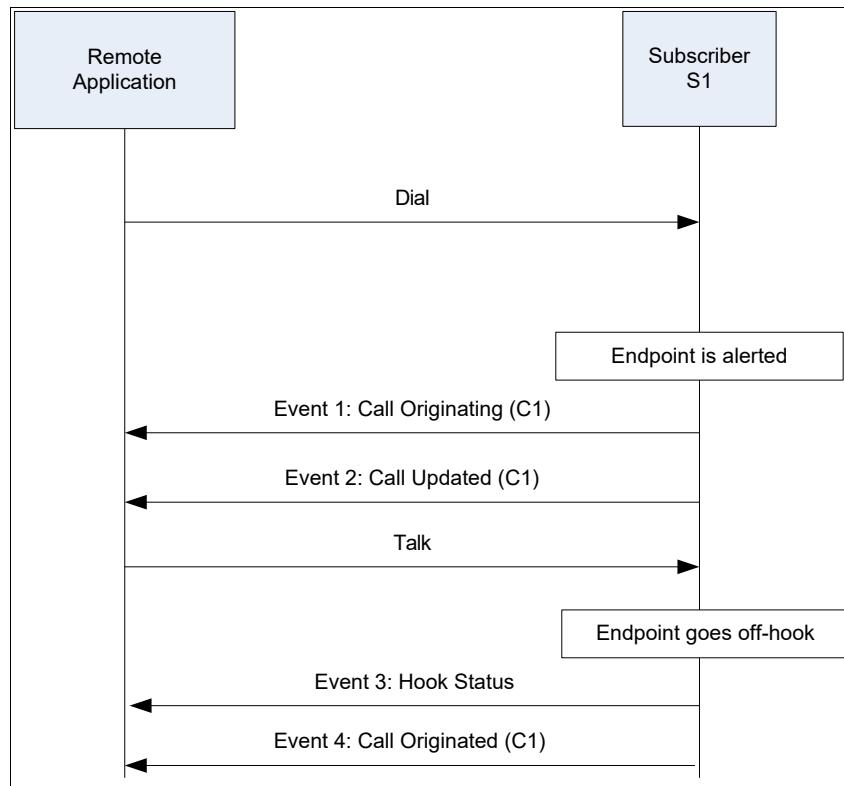


Figure 30 Call Answered Through Talk Request

The same *Call Originating* event is generated as in the previous subsection, but this time followed by a *Call Updated* event. The *Call Updated* event contains an *allowAnswer* parameters specifying that the call can be answered by a remote application.

The remote application sends a *Talk* request to answer the call on behalf of the subscriber. A *Hook Status* and a *Call Originated* are sent once the endpoint has gone off-hook.

Call automatically answered

If the subscriber's endpoint supports the *answer-after* parameter of the *SIP Call-Info* header and if Cisco BroadWorks is configured accordingly, then the endpoint is automatically forced to go off-hook when performing a Dial request. The following diagram illustrates the request and events exchanged.

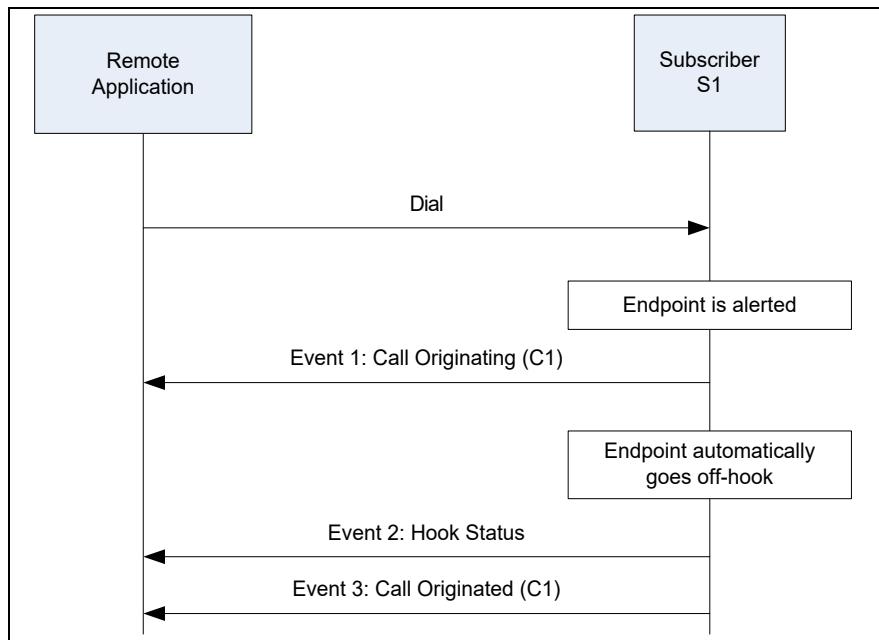


Figure 31 Call Answered Through Talk Request

5.2.1.6.2 Case 2

In this case, the same events as described for Case 1 are produced along with the following: A *Call Held* event is generated against subscriber S1 (call C1) and subscriber S2 (call C2).

More information on events as well as their content is provided in section [6.2 Call Events](#).

5.2.1.7 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111000	Invalid location value <location>.
400	111001	Cannot find location <locationAddress>.
400	111002	The locationAddress not allowed for location <location>.
400	111003	User is not assigned the <service name> service or there is no location assigned to this service.
400	111004	User does not have a primary endpoint.
400	110820	Blocked by Remote Office.
400	100006	Endpoint state is not valid.
400	110581	Emergency call is present.
400	110582	Blocked by prepaid service.

statusCode	errorCode	Summary
400	110580	User has no charge address.
400	110646	Not allowed to have more than one call outside the conference.
400	100019	Blocked by maximum simultaneous calls policy.
400	110011	Answer confirmation in progress.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Address errors, see section 5.1.3.2 Address Error .		

5.2.2 Talk

5.2.2.1 Description

This request is used by a remote application to answer an unanswered call or retrieve a held call.

A Talk request can be used to answer an unanswered call if the subscriber's endpoint supports the remote control talk event package. For more information, see the *Cisco BroadWorks SIP Access Side Extensions Interface* [8].

A Talk request can be used to retrieve a held call if the call was previously held using the Hold request. If the call was held from the endpoint, then the Talk request can only be used to retrieve the held call if the endpoint supports the remote control talk event package. For more information, see the *Cisco BroadWorks SIP Access Side Extensions Interface* [8].

This means that a Talk request can only be executed in the following conditions:

- The call is in state *Alerting* and has a “Terminator” or “Click-to-Dial” personality.
- The call is in state *Held*.

The following diagrams provide information on the state and topology changes involved.

5.2.2.1.1 Case 1: Talk on Incoming Alerting Call

Pre-conditions: Subscriber S1 is calling subscriber S2, both calls are in the alerting state. Subscriber S2 endpoint supports the remote control talk event package.

Request: The remote application sends a Talk request for call C2 to answer the incoming call.

Post-conditions: The call is answered and both calls are in the Active state.

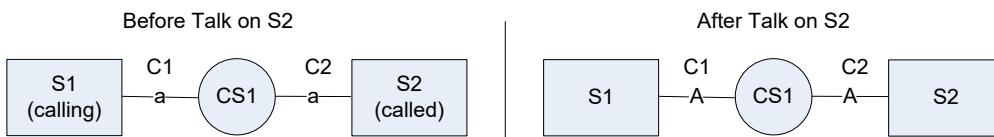


Figure 32 Talk on Incoming Alerting Call

5.2.2.1.2 Case 2: Talk on Held Call (Held via Hold Request)

Pre-conditions: The subscriber S1 is actively connected to subscriber S2. The subscriber S1 puts the call on hold. Call C1 is in the held state and Call C2 is in the Remote Hold state.

Request: The remote application sends a Talk request for call C1 to retrieve the call from hold.

Post-conditions: The calls C1 and C2 are put back in the Active state.

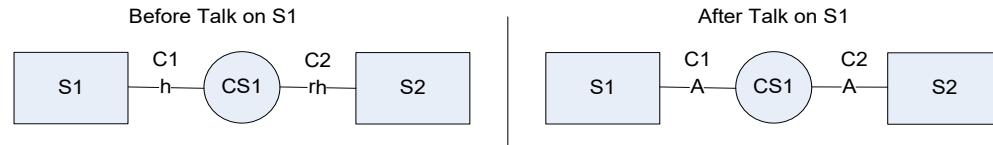


Figure 33 Talk on Held Call (Held via Hold Request)

5.2.2.1.3 Case 3: Talk on Held Call (Held via Endpoint)

Pre-conditions: The subscriber S1 is actively connected to subscriber S2. The subscriber S1 puts the call on hold. Call C1 is in the held state and Call C2 is in the Remote Hold state. Subscriber S2 endpoint supports the remote control talk event package.

Request: The remote application sends a Talk request for call C1 to retrieve the call from hold.

Post-conditions: The calls C1 and C2 are put back in the Active state.

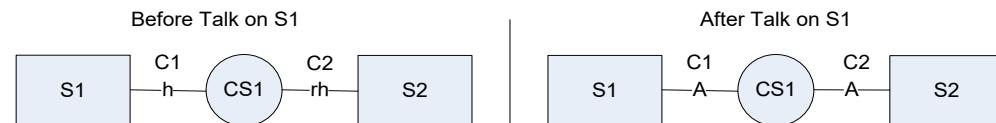


Figure 34 Talk on Held Call (Held via Endpoint)

5.2.2.1.4 Case 4: Talk on “Click-to-Dial” Incoming Call

Pre-conditions: The remote application sends a dial request for subscriber S1 with S2 address. The call C1 goes into alerting and the subscriber S1 is prompted to go-off-hook. Subscriber S1 endpoint supports the remote control talk event package.

Request: The remote application sends a Talk request for call C1 to answer the incoming call.

Post-conditions: The call C1 is answered so its personality goes from Click-to-Dial to Originator and the call C2 to S2 is initiated.

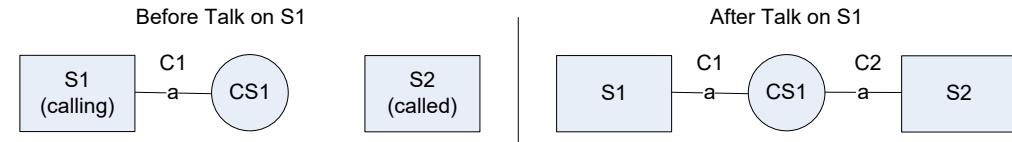


Figure 35 Talk on Click-To-Dial Incoming Call

5.2.2.1.5 Case 5: Talk on Held Call with One Active Call

Pre-conditions: The subscriber S1 is connected to subscriber S3 through call C3 and to subscriber S2 through call C1. Call C1 is the held state, call C3 is in the Active or Remote Hold state. Subscriber S1 endpoint supports the ability to retrieve a call through third party call control.

Request: The remote application sends a Talk request for call C1.

Post-conditions: The calls C1 and C2 are put back in the Active state. Call C3 is put on hold.

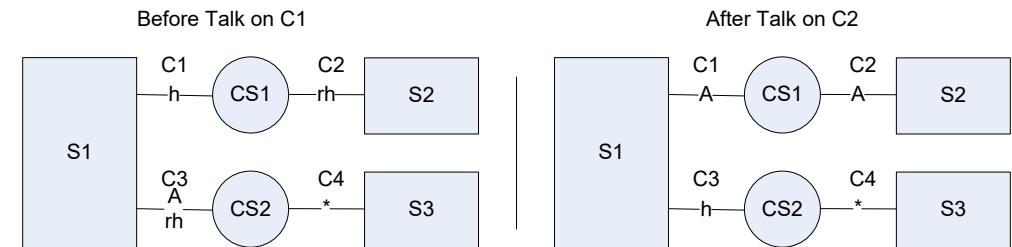


Figure 36 Talk on Held Call with One Active Call

5.2.2.1.6 Case 6: Talk on an Held Call with One Alerting Call (Personality Terminator)

Pre-conditions: The subscriber S1 is connected to subscriber S3 through call C3 and to subscriber S2 through call C1. Call C1 is the held state, call C3 is in the Alerting state with personality Terminator. Subscriber S1 endpoint supports the ability to retrieve a call through third party call control.

Request: The remote application sends a Talk request for call C1.

Post-conditions: The calls C1 and C2 are put back in the Active state. Call C3 remains alerting.

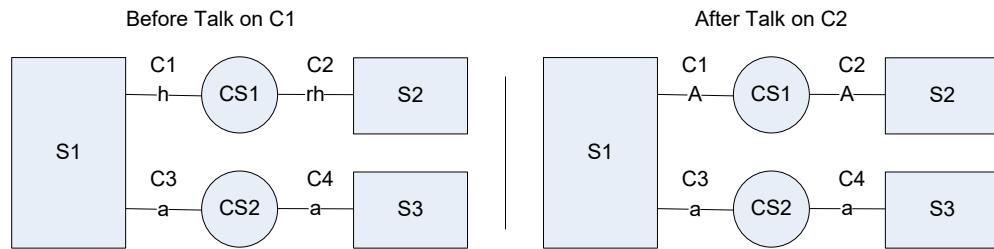


Figure 37 Talk on Held Call with One Alerting Call (Personality Terminator)

5.2.2.1.7 Case 7: Talk on Held Call with One Alerting Call (Personality Originator)

Pre-conditions: The subscriber S1 is connected to subscriber S3 through call C3 and to subscriber S2 through call C1. Call C1 is the held state, call C3 is in the Alerting state with personality Originator. Subscriber S1 endpoint supports the ability to retrieve a call through third party call control.

Request: The remote application sends a Talk request for call C1.

Post-conditions: The calls C1 and C2 are put back in the Active state. Call C3 and C4 are released.

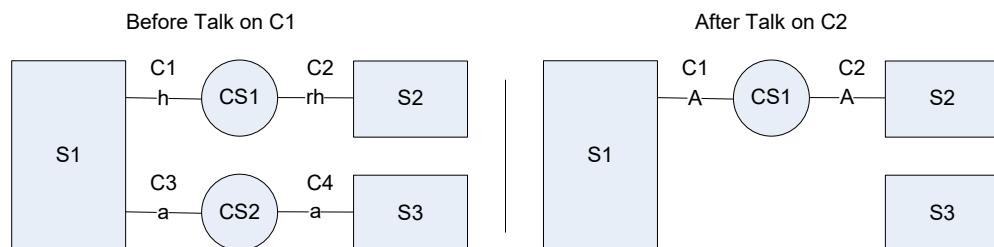


Figure 38 Talk on Held Call with One Alerting Call (Personality Originator)

5.2.2.1.8 Case 8: Talk on Held Call with One Alerting Call (Personality Originator) Established Through Dial Request

Pre-conditions: The subscriber S1 is connected to subscriber S3 through call C3 and to subscriber S2 through call C1. Call C1 is the held state, call C3 is in the Alerting state with personality Originator. Call C3 was established through a Dial request. Subscriber S1 endpoint supports the ability to retrieve a call through third party call control.

Request: The remote application sends a Talk request for call C1.

Post-conditions: The calls C1 and C2 are put back in the Active state. Call C3 is in the held state. The media received by the endpoint (that is, the device) associated with call C3 must ignore any media that might be received.

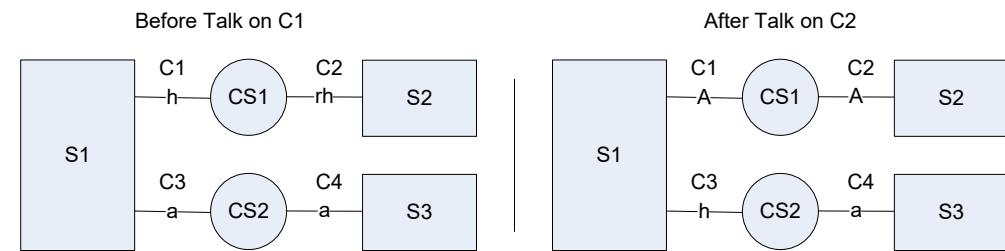


Figure 39 Talk on Held Call with One Alerting Call (Personality Originator) Established Through Dial Request

5.2.2.1.9 Case 9: Talk on Held Executive Call

Pre-conditions: A call session CS1 is established between subscriber S1 (assistant) and subscriber S3. The call was established as the result of an origination by the assistant on behalf of the executive (S2), or as the result of the routing of a terminating executive call to the assistant. Call C1 is in the Remote Hold state, C2 is in the Held state, C3 is in the Remote Hold state.

Request: The remote application sends a Talk request for call C2.

Post-conditions: All calls go in the Active state.

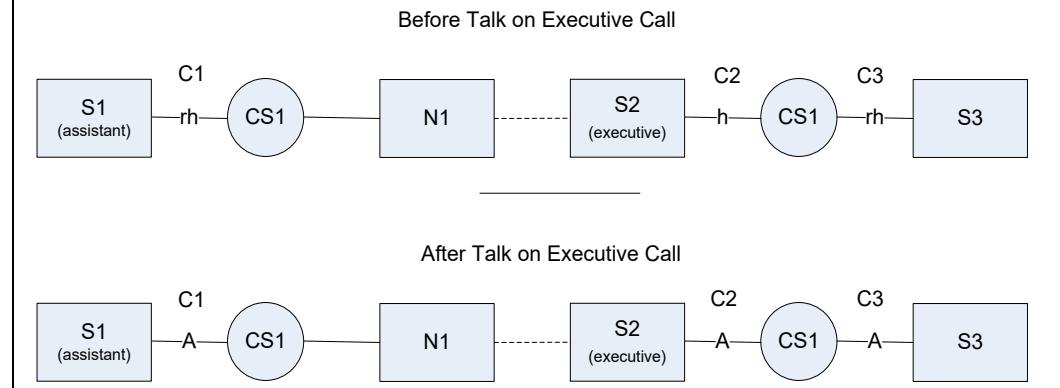


Figure 40 Talk on Executive's Call

5.2.2.2 Request

A remote application can issue this request by sending a Talk Request. A Talk Request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID associated with the call to answer or retrieve.
<i>callId</i>	CallId	Yes	The call ID of the call to answer or retrieve.

5.2.2.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.2.2.4 Associated Events

The ability to handle a Talk request (that is, to support third-party call control answer) is advertised by a *Call Updated* event through the *allowAnswer* parameter. For more information, see section [8 Type Definitions](#). This event is produced when a call enters the *Alerting* state.

When the talk request is accepted, the following events are generated:

- In case 1, a *Call Answered* event is generated for both calls (C1 and C2) indicating that the call has been answered on behalf of the subscriber.
- In case 2 and case 3, a *Call Retrieved* event is generated for call C1 and C2.
- In case 4, a *Call Originated* is generated for call C1.
- In case 5, a *Call Answered* event is generated for both calls C1 and C2, a *Call Held* is generated for call C3, a *Call Held* is generated for call C4 only if it was not already held by S3.
- In case 6, a *Call Retrieved* event is generated for both calls C1 and C2.
- In case 7, a *Call Retrieved* event is generated for both calls C1 and C2 and a *Call Released* for both calls C3 and C4.
- In case 8, a *Call Retrieved* event is generated for both calls C1 and C2 and a *Call Held* for calls C3.
- In case 9, a *Call Retrieved* event is generated for calls C1, C2, and C3.

Note that in the case where S2 and S3 were not Cisco BroadWorks subscribers (that is, are just Network Parties) then no event would be generated against them and their associated call (C2 and C4).

5.2.2.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110581	Emergency call is present.
400	100020	No endpoint for call <callId>.
400	110620	Call <callId> is alerting but answer is not supported by the device or endpoint.

statusCode	errorCode	Summary
400	101001	State for call <callId> is not valid.
400	100021	Endpoint state for call <callId> is not valid.
400	100006	Endpoint state is not valid.
400	110501	Consultation transfer in progress.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.2.3 Hold

5.2.3.1 Description

This request is used by a remote application to put a call on hold. As a result, the call goes into Hold and the remote subscriber call state goes into *Remote Hold*.

The Hold request can only be performed if the call is in one of the following states:

- *Active*
- *Remote Hold*

The following two diagrams provide information on the state and topology changes involved.

5.2.3.1.1 Case 1: Hold an Active Call

Pre-conditions: Subscriber S1 and subscriber S2 are in Active state.

Request: The remote application sends a Hold request for call C1.

Post-conditions: Call C1 is in Held state, call C2 is in the Remote Hold state

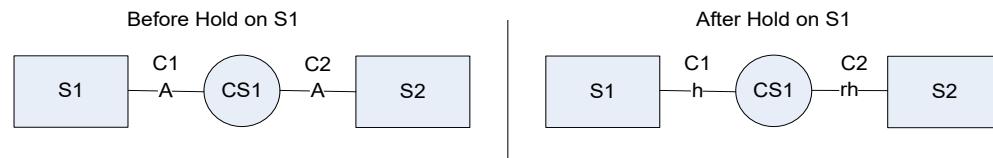


Figure 41 Hold Active Call

5.2.3.1.2 Case 2: Hold Remote Held Call

Pre-conditions: Subscriber S2 has put subscriber S1 on hold. So S1 is in Remote Held state and S2 is in the Held state.

Request: The remote application sends a Hold request for call C1.

Post-conditions: Call C1 is in Held state, call C2 remains in the Held state.

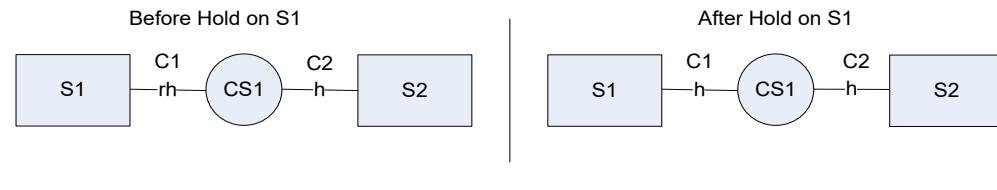


Figure 42 Hold Remote Held Call

5.2.3.1.3 Case 3: Hold Executive Call

When an executive has an answered call with an assistant location, there are special interactions when either the assistant or the executive holds the call.

If the assistant holds the call, the executive receives the hold request from the assistant location and the state for both the assistant and the executive's calls.

If the executive holds the call, then the state for the executive call is held but the state for the assistant is *Remote Held* since a hold request from the executive and a hold request from the remote party are identical from the assistant's perspective.

Pre-conditions: A call session CS1 is established between subscriber S1 (assistant) and subscriber S3. The call was established as the result of an origination by the assistant on behalf of the executive (S2), or as the result of the routing of a terminating executive call to the assistant.

Request: The remote application sends a Hold request for call C2.

Post-conditions: Call C1 goes in the Remote Held state if it was Active, call C2 goes in the Held state.

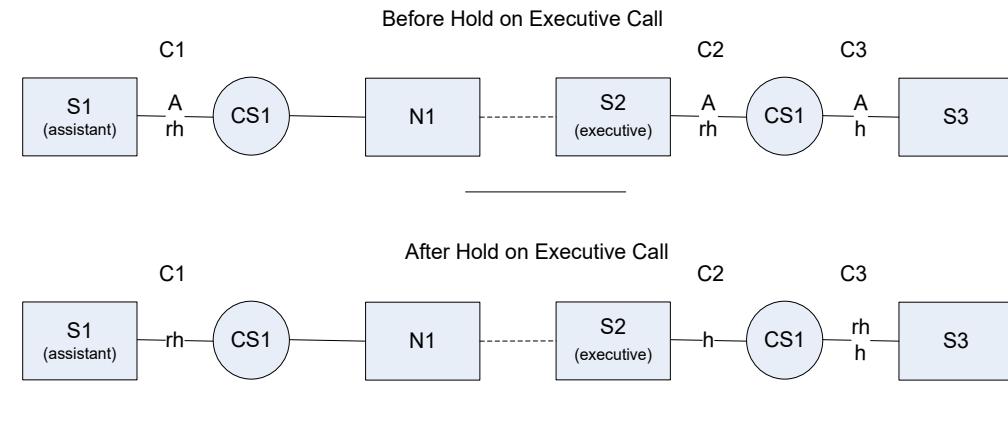


Figure 43 Executive's Hold

5.2.3.2 Request

A remote application can issue this request by sending a Hold Request. A Hold Request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID associated with the call to put on hold.
<i>callId</i>	CallId	Yes	The call ID of the call to put on hold.

5.2.3.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.2.3.4 Associated Events

In the first two cases, when the Hold request is accepted, a *Call Held* is generated for call C1 and call C2. In the Executive-Assistant case, a *Call Held* is also generated for call C3.

5.2.3.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	101003	Call <callId> is an Emergency call.
400	100021	Endpoint state for call <callId> is not valid.
400	100020	No endpoint for call <callId>.
400	101001	State for call <callId> is not valid.
400	110010	Answer confirmation in progress for call <callId>.
400	100006	Endpoint state is not valid.
400	110501	Consultation transfer in progress.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.2.4 Release

5.2.4.1 Description

This request is used by a remote application to release a call. A call can be released when it is in any state except *Idle* or *Detached*.

If a subscriber is part of a conference, then only the subscriber call will be released from the conference; the other participants will remain in the conference.

The following diagrams provide information on the state and topology changes involved.

5.2.4.1.1 Case 1: *Release a Call*

Pre-conditions: Subscriber S1 and subscriber S2 are in any state except Idle or Detached.

Request: The remote application sends a Release request for call C1.

Post-conditions: Call C1 and C2 are released as well as the Call Session CS1.

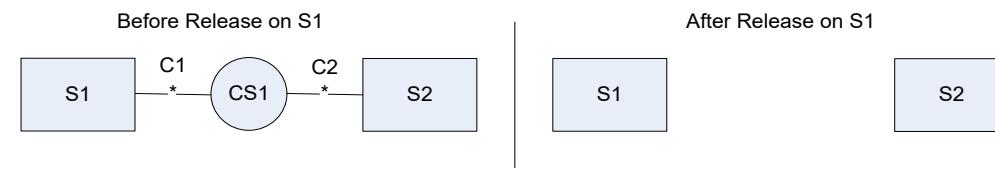


Figure 44 Release Call

5.2.4.1.2 Case 2: *Release an Alerting Call without Decline Option*

Pre-conditions: Subscriber S1 and subscriber S2 are in any state except Idle or Detached.

Request: The remote application sends a Release request for call C1.

Post-conditions: Call C1 and C2 are released as well as the Call Session CS1.

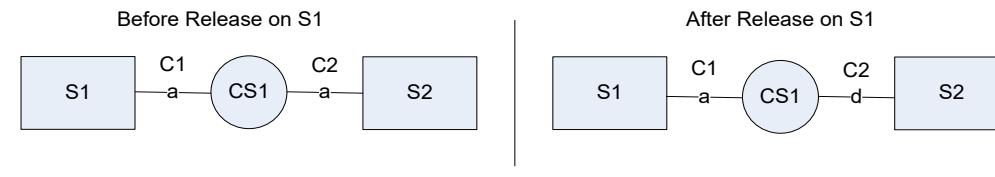


Figure 45 Release Call without Decline Option

5.2.4.1.3 Case 3: Release Call Participating in a Conference (Endpoint Initiated Cisco BroadWorks Conference)

Pre-conditions: Subscriber S2 is the controller of a conference with subscriber S1 and S3. The conference is an endpoint initiated BroadWorks conference.

Request: The remote application sends a Release request for call C1.

Post-conditions: Call C1 and C2 are released as well as the Call Session CS1. The conference is maintained however since this type of conference can have zero or more participant.

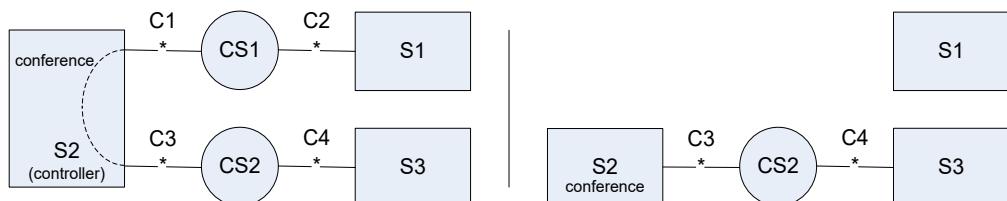


Figure 46 Release Call Participating in Conference (Endpoint-Initiated Cisco BroadWorks Conference)

5.2.4.1.4 Case 4: Release Call Participating in Conference (Cisco BroadWorks Conference)

Pre-conditions: Subscriber S2 is the controller of a conference with subscriber S1 and S3. The conference is a BroadWorks conference.

Request: The remote application sends a Release request for call C1.

Post-conditions: Call C1 and C2 are released as well as the Call Session CS1. The conference is released because a BroadWorks conference requires two or more participant. However, the call session remains between subscriber S1 and S3.

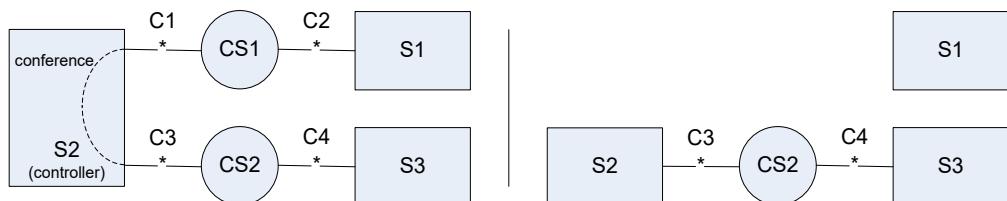


Figure 47 Release Call Participating in Conference (Cisco BroadWorks Conference)

5.2.4.1.5 Case 5: Release Call Participating in Conference (Three Participants)

Pre-conditions: Subscriber S2 is the controller of a conference with subscriber S1, S3 and S4.

Request: The remote application sends a Release request for call C1.

Post-conditions: Call C1 and C2 are released as well as the Call Session CS1. The conference is not released as call session CS2 and CS3 remain.

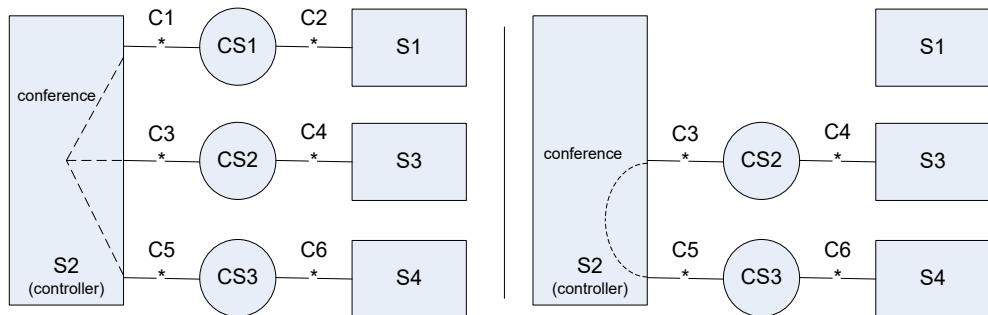


Figure 48 Release Call Participating in Conference (Three Participants)

5.2.4.1.6 Case 6: Release Executive Call

When an executive has an answered call with an assistant location, the executive can release the call.

Pre-conditions: A call session CS1 is established between subscriber S1 (assistant) and subscriber S3. The call was established as the result of an origination by the assistant on behalf of the executive (S2), or as the result of the routing of a terminating executive call to the assistant.

Request: The remote application sends a Release request for call C2.

Post-conditions: Call C1, C2, and C3 are released.

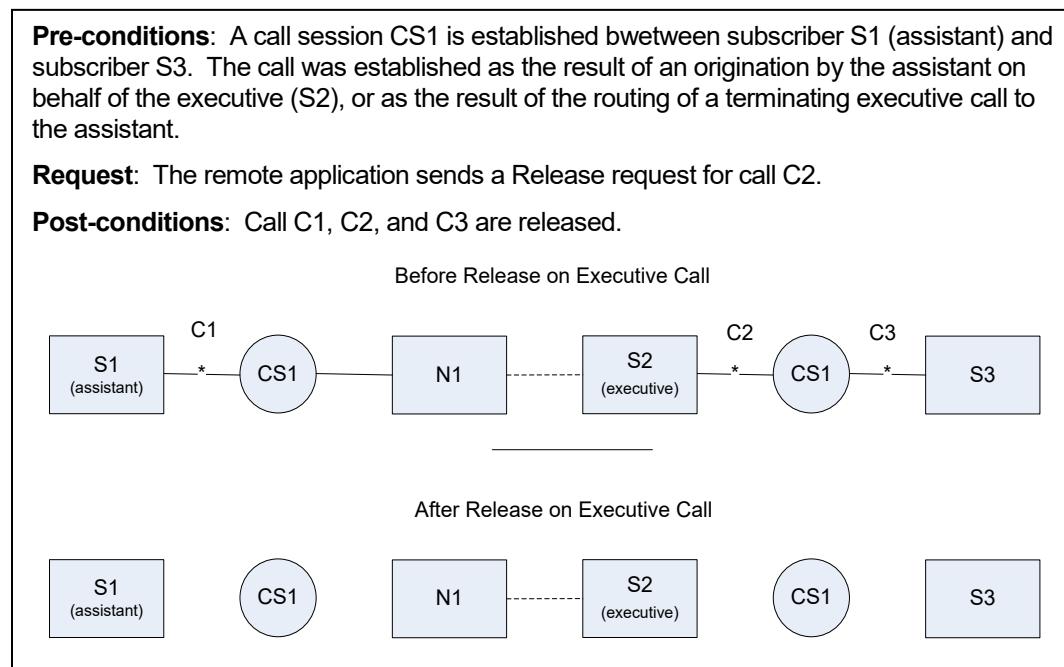


Figure 49 Executive's Release

5.2.4.2 Request

A remote application can issue this request by sending a Release Request. A Release Request contains the common parameters listed in section [5.1.2 Response](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID associated with the call to release.
<i>callId</i>	CallId	Yes	The call ID of the call to release.
<i>decline</i>	EmptyContent	No	<p>The decline element only applies when the specified call is in the <i>Alerting</i> state with Terminator personality (an unanswered, incoming call). For all other types of call, the decline element is ignored.</p> <p>When a Release Request is received for a call in the <i>Alerting</i> state with Terminator personality and the decline element is present, then the call is immediately sent to <i>Temporarily Unavailable</i> processing, which can trigger no answer services such as Voice Mail and Call Forwarding No Answer.</p> <p>When a Release Request is received for a call in the <i>Alerting</i> state with Terminator personality and the decline element is not present, then the call is “ignored”. The call is detached so it is no longer visible and the user’s device(s) stops alerting for the call, but the call is not immediately sent to <i>Temporarily Unavailable</i> processing. Instead, the caller continues to hear ringback and the user’s own no answer services (such as Voice Mail and Call Forwarding No Answer) continue to run until their No Answer timer expires or the call is released in some other way (such as by the caller hanging up).</p>

5.2.4.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.2.4.4 Associated Events

When the Release request is accepted, the following events are generated:

- In case 1, when the Release request is accepted a *Call Released* event is generated for both calls (C1 and C2). A *Hook Status* event is generated for both users.
- In case 2, when the Release request is accepted a *Call Detached* event is generated for call C2. A *Hook Status* event is generated for subscriber S2.
- In cases 3 and 4, when the Release request is accepted, a *Conference Released* event is generated against subscriber S2 and a *Call Released* event is generated for call C1. A *Hook Status* event is generated for subscriber S1.
- In case 5, when the Release request is accepted, a *Call Released* event is generated for all calls (C1, C2, and C3). A *Hook Status* event is generated for all users.

5.2.4.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	101003	Call <callId> is an Emergency call.
400	101001	State for call <callId> is not valid.
400	110501	Consultation transfer in progress.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Address errors, see section 5.1.3.2 Address Error .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.2.5 Blind Transfer

5.2.5.1 Description

This request is used to transfer an existing call to a new destination. This transfer is performed in a single step, that is, the client application doing the transfer does not have to place the existing call on hold before issuing the transfer. The subscriber must have the Call Transfer service to use this request.

A call can be transferred when it is in the *Alerting*, *Held*, *Remote Held*, or *Active* state. When it is in the *Alerting* state, it must be have its personality set to “Terminator”.

The following diagram provides information on the state and topology changes involved.

5.2.5.1.1 Case 1: Basic Blind Transfer

Pre-conditions: Call C1 can be in the Held, Remote Held, and Alerting or Active state.
Request: The remote application sends a Blind Transfer specifying call C1 and S3 phone number.
Post-conditions: Call C1 is released and goes into Detached. Call Session CS1 is maintained. Call C3 is created. Both C2 and C3 are in the Alerting state.

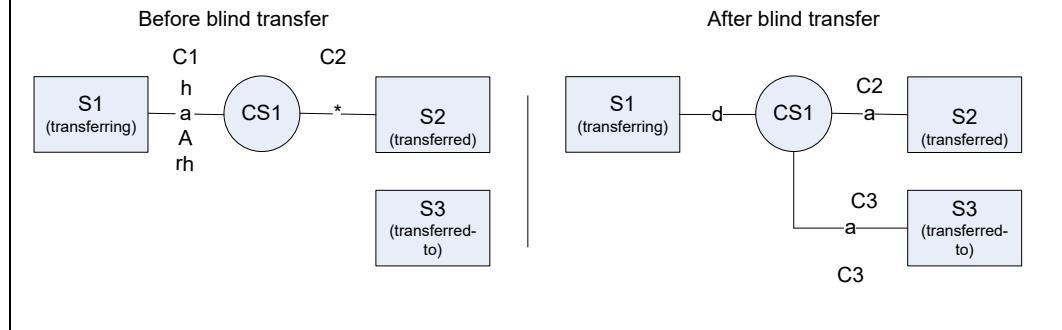


Figure 50 Blind Transfer State Changes

5.2.5.1.2 Case 2: Executive Blind Transfer

Pre-conditions: A call session CS1 is established between subscriber S1 (assistant) and subscriber S3. The call was established as the result of an origination by the assistant on behalf of the executive (S2), or as the result of the routing of a terminating executive call to the assistant.

Request: The remote application sends a Blind Transfer specifying call C2 and S4 phone number.

Post-conditions: Call C1 is released. Call C2 is released and goes into detached. Call Session CS1 is maintained. Call C4 is created. Both C3 and C4 are in the Alerting state.

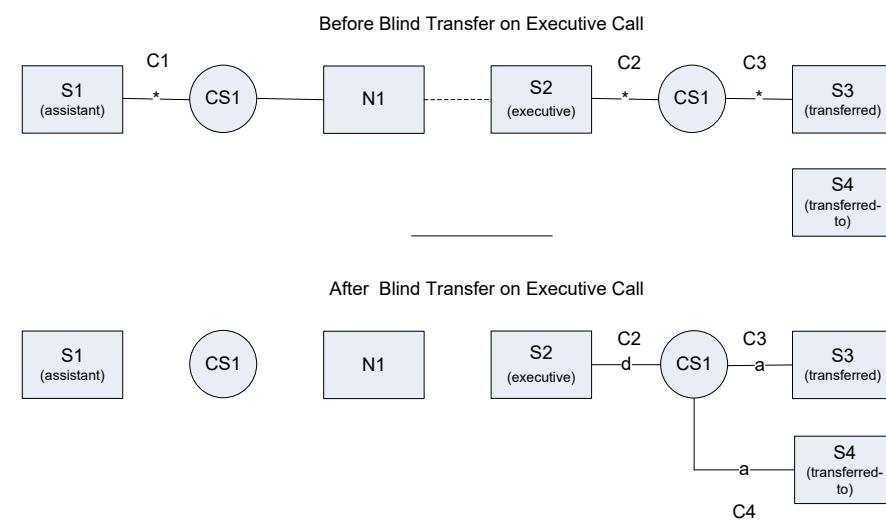


Figure 51 Executive Blind Transfer State Changes

5.2.5.2 Request

A remote application can issue this request by sending a *BlindTransferRequest*. A *BlindTransferRequest* contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID associated with the call to transfer.
<i>callId</i>	CallId	Yes	The call ID of the call to transfer.
<i>address</i>	Address	Yes	The destination address for the transfer.

5.2.5.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).



5.2.5.4 Associated Events

5.2.5.4.1 Case 1

When a Blind Transfer request is accepted, the following events are generated:

- A *Call Redirected* event against the transferring subscriber's call (C1 in the diagram).
- A *Call Transferred* event against the transferred subscriber's call (C2 in the diagram).
- A *Call Received* event against the transferred-to subscriber's call (C3 in the diagram).
- A *Hook Status* event against the transferring subscriber's call (assuming the subscriber is not being active on any other call).

If the called destination is not available (for example, busy) then the following events are generated:

- A *Call Redirected* event against the transferring subscriber's call (C1 in the diagram).
- A *Call Releasing* event with a releaseCause set to "busy" against the transferred subscriber's call (C2 in the diagram) followed by a *Call Released* event.
- A *Call Received* against the transferred-to subscriber's call (C3 in the diagram) followed by a *Call Released* event with a ReleaseCause set to "busy".

Note the following about subscriber S2 and S3:

- If S2 and S3 were not Cisco BroadWorks subscribers (that is, are just Network Parties) then no event would be generated against them and their associated call (C2 and C3).
- If S2 was a Route Point then a *Route Point Call Updated* event would be generated in place of the *Call Transferred* event.
- If S2 was an ACD the *ACD Call Updated* event would be generated in place of the *Call Transferred* event.

5.2.5.4.2 Case 2

When a Blind Transfer request is accepted, the following events are generated:

- A *Call Redirected* event against the transferring subscriber's call (C2 in the diagram).
- A *Call Transferred* event against the transferred subscriber's call (C3 in the diagram).
- A *Call Received* event against the transferred-to subscriber's call (C4 in the diagram).
- A *Hook Status* event against the transferring subscriber's call (assuming the subscriber is not active on any other call).
- A *Call Released* event against the assistant's call (C1 in the diagram).
- A *Hook Status* event against the assistant's call (assuming the subscriber is not active on any other call).

5.2.5.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	101003	Call <callId> is an Emergency call.
400	110500	User is not assigned the Call Transfer service.
400	101005	Diversions are inhibited for call <callId>.



statusCode	errorCode	Summary
400	101001	State for call <callId> is not valid.
400	101000	Call <callId> has no remote call.
400	110541	Call <callId> is an answered Push-To-Talk call.
400	100008	Remote party for call <callId> is a virtual subscriber.
400	100017	Blocked by maximum concurrent redirections policy.
401	100016	Blocked by Account/Authorization Codes service.
400	100018	Blocked by Intercept Group or Intercept User service.
400	100007	Blocked by Outgoing Calling Plan service.
400	110501	Consultation transfer in progress.
400	100009	Blocked by Communication Barring service.
400	100010	Blocked by translations.
400	110208	Transfer target must be a Call Center when using transfer to front function.
400	110204	User is not an agent for the call center.
400	110507	Call <callId> with recall type <recallType> cannot be redirected prior to answer.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Address errors, see section 5.1.3.2 Address Error .		

5.2.6 Consultative Transfer

5.2.6.1 Description

This request is used to transfer a specified call after consulting with the receiving party. The Consultative Transfer is also known as an Attended Transfer, or a Transfer with Consultation. The subscriber must have the Call Transfer service to use this request.

From a call topology point of view, the Consultative Transfer merges two call sessions into a new one. All calls are maintained but are connected to the new call session.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: Call C1, C2, C3 and C4 can be in the Held, Remote Held, Alerting or Active state. When C2 or C3 are in the Alerting state, it must be have its personality set to Originator. C2 and C3 cannot be both in the Alerting state.

Request: The remote application sends a Consultative Transfer request. BroadWorks merges CS1 and CS2 into a new Call Session CS3 with a new externalTrackingId. It also determines the new call personality ("Terminator" or "Originator") for C1 and C4.

Post-conditions: C2 and C3 go into Detached. C1 and C4 states are set to the appropriate value. See state transition table for more details.

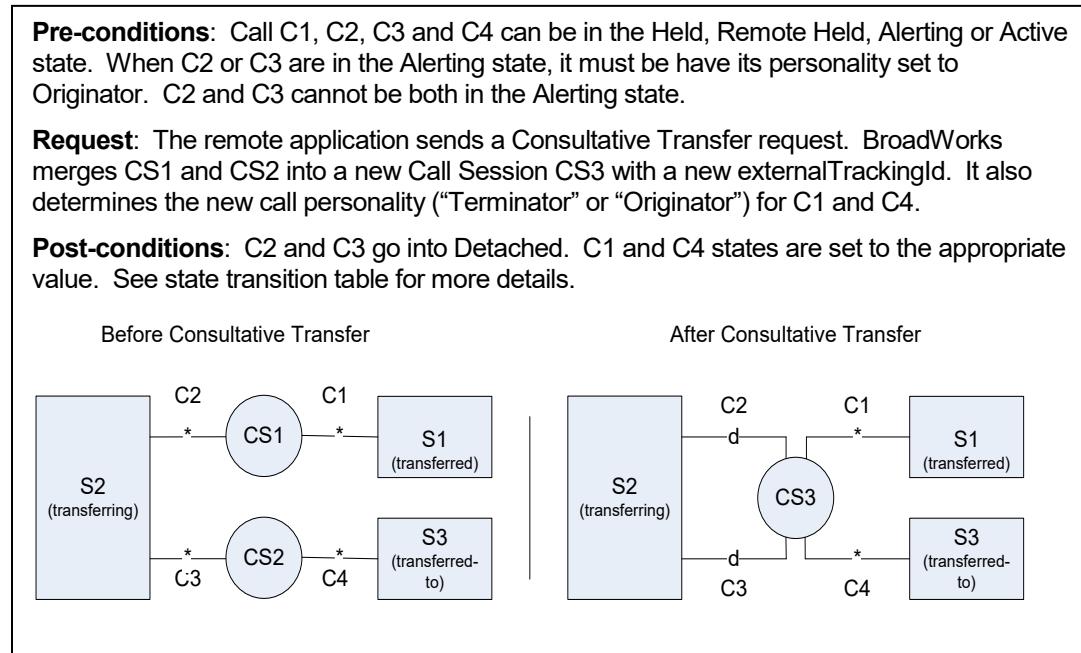


Figure 52 Consultative Transfer State Changes

As mentioned in the diagram, the personality of each call may be adjusted to ensure that there is one originator and one terminator personality assigned to each active call.

The following table describes how call C1 and C4 state and personality is affected by the Consultative Transfer request, assuming that C1 is the first call specified in the request. In general, the following rules are applied (in order):

- 1) A call that is in state *Alerting* with personality terminator remains with that state and personality.
- 2) If both calls are in the *Active* state, then the first call specified in the request keeps its current personality.
- 3) If only one call is in the *Active* state, then its personality is changed to originator.
- 4) In other cases, the first call specified in the request keeps its current personality.

Pre-transfer State		Post-transfer State	
C1 State Personality	C4 State Personality	C1 State Personality	C4 State Personality
ActiveOriginator/ Terminator	AlertingTerminator	AlertingOriginator	AlertingTerminator
HeldOriginator/ Terminator	AlertingTerminator	AlertingOriginator	Alerting Terminator
RemoteHeldOriginator/ Terminator	AlertingTerminator	AlertingOriginator	AlertingTerminator
AlertingTerminator	ActiveOriginator/ Terminator	AlertingTerminator	AlertingOriginator

Pre-transfer State		Post-transfer State	
C1 State Personality	C4 State Personality	C1 State Personality	C4 State Personality
AlertingTerminator	HeldOriginator/ Terminator	AlertingTerminator	AlertingOriginator
AlertingTerminator	RemoteHeldOriginator/ Terminator	AlertingTerminator	AlertingOriginator
ActiveOriginator/ Terminator	HeldOriginator/ Terminator	RemoteHeldOriginator	HeldTerminator
ActiveOriginator/ Terminator	RemoteHeldOriginator/ Terminator	ActiveOriginator	ActiveTerminator
ActiveOriginator	ActiveOriginator/ Terminator	ActiveOriginator	ActiveTerminator
ActiveTerminator	ActiveOriginator/ Terminator	ActiveTerminator	ActiveOriginator
HeldOriginator/ Terminator	ActiveOriginator/ Terminator	HeldTerminator	Remote HeldOriginator
RemoteHeldOriginator/ Terminator	ActiveOriginator/ Terminator	ActiveTerminator	ActiveOriginator
HeldOriginator	HeldOriginator/ Terminator	HeldOriginator	HeldTerminator
HeldTerminator	RemoteHeldOriginator/ Terminator	HeldTerminator	Remote HeldOriginator
RemoteHeldOriginator	HeldOriginator/ Terminator	RemoteHeldOriginator	HeldTerminator
RemoteHeldTerminator	RemoteHeldOriginator/ Terminator	ActiveTerminator	ActiveOriginator

5.2.6.2 Request

A remote application can issue this request by sending a ConsultativeTransferRequest. A ConsultativeTransferRequest contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID associated with the call to transfer.
<i>callId1</i>	CallId	Yes	The call ID of the call to transfer.
<i>callId2</i>	CallId	Yes	The call ID of the consulted call.

5.2.6.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.2.6.4 Associated Events

When a Consultative Transfer request is accepted, the following events are generated:

- A *Call Redirected* event against the transferring subscriber's call (C2 and C3 in the diagram).
- A *Call Transferred* event against the transferred subscriber's call (C1 in the diagram).
- A *Call Transferred* event against the transferred-to subscriber's call (C4 in the diagram).
- A *Hook Status* event against the transferring subscriber's call (S2 in the diagram).

Each event contains the external tracking ID of the new call sessions.

Note the following about subscriber S1 and S3:

- If S1 and S3 were not Cisco BroadWorks subscribers (that is, are just Network Parties) then no event would be generated against them and their associated call (C1 and C4).
- If S1 (or S3) was a Route Point then a *Route Point Call Updated* event would be generated in place of the *Call Transferred* event.
- If S1 (or S3) was an ACD then an *ACD Call Updated* event would be generated in place of the *Call Transferred* event.

5.2.6.4.1 Impact on External Tracking ID

The external tracking ID for each call changes on the transfer with consultation action.

This change is necessary as the transferring party has two independent calls each with their own unique call sessions and external tracking ID to two other parties. The transfer function on Cisco BroadWorks is independent of the call topology. This means that a transfer action can be generated on any two calls regardless of the direction of the calls (for example, incoming to the transferring party, outgoing from the transferring party) and regardless of whether the transferring party placed a consultation call or received an incoming call.

The act of the transfer connects the calls of the two parties together where each party has a different external tracking ID. As a result of this action, a new external tracking ID is generated and the transferred calls are updated with this tracking ID.

The transferred calls, C1 for party S1 and C4 for party S3 are updated via a *Call Transferred* event for each monitored party. The *Call Transferred* event will always carry the new external tracking ID for the call associated to the monitored party. The call ID in the *Call Transferred* event remains the same. The external tracking ID, the remote party, and the personality are updated for this scenario.

The transferring party's calls, C2 and C3 for party S2 are updated via a *Call Redirected* event with redirecting information for both calls being transferred together. The external tracking ID is updated in each call in the *Call Redirected* event. The call state is updated to detached state for each call. A call redirect is included with a redirection reason of transfer for each call. The call ID for calls C2 and C3, respectively, remains unchanged.

The Cisco BroadWorks CTI interface does not allow the Route Point to perform a transfer with consultation action so the Route Point can never be a transferring party. However, the Route Point can be a transferred party. In this case, the Route Point is updated with a *Route Point Call Updated* event indicating that the Route Point queue entry has been transferred and the updated external tracking ID.



5.2.6.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	101007	Call <callId> does have the expected session ID of <sessionId>.
400	110500	User is not assigned the Call Transfer service.
400	101008	Invalid call IDs.
400	100022	Blocked by Zone Calling Restrictions.
400	110010	Answer confirmation in progress for call <callId>.
400	101000	Call <callId> has no remote call.
400	101003	Call <callId> is an Emergency call.
400	110540	The call <callId> is a Push-To-Talk call.
400	101001	State for call <callId> is not valid.
400	110505	Calls are not in proper state. Call states cannot be Alerting for both calls.
400	110504	Remote party for call <callId> cannot be transferred while the transfer target is Alerting.
400	100009	Blocked by Communication Barring service.
400	100007	Blocked by Outgoing Calling Plan service.
400	100023	Blocked by Incoming Calling Plan service.
400	110502	Transfer has failed.
400	110501	Consultation transfer in progress.
400	100006	Endpoint state is not valid.
400	110503	Consultative transfer already in progress.
400	100017	Blocked by maximum concurrent redirections policy.
400	100018	Blocked by Intercept Group or Intercept User service.
400	110110	Emergency escalation in progress.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.2.7 Mute Transfer

5.2.7.1 Description

This request is used to transfer an existing call to a new destination. The remote application doing the transfer does not have to establish a call to the transfer destination or place the existing call on hold before issuing the transfer. The transfer request is considered successful if the transfer destination begins ringing. The subscriber must have the Call Transfer service to use this request.

The Mute Transfer can be seen as an automated version of the Consultative Transfer where the establishment of the second call session to the transfer destination is done automatically by Cisco BroadWorks. If the remote destination rings then the two call sessions are merged into a new one. If the transfer destination does not ring then the transfer request is aborted and the call topology remains the same.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: Call C1, and C2 can be in the Held, Remote Held, or Active state.

Request: The remote application sends a Mute Transfer request specifying subscriber S1call C1 and the phone number of S3. BroadWorks establishes a call C3, a call session CS2 (not shown), and a call C4 to subscriber S3. When subscriber S3 device starts to ring, BroadWorks merges CS1 and CS2 into a new call session CS3 with a new externalTrackingId. C4 personality remains set to Terminator. If C2 personality is not set "Originator", it is changed to "Originator".

Post-conditions: C1 and C3 go in to the Detached state. C2 and C4 are in the Alerting state.

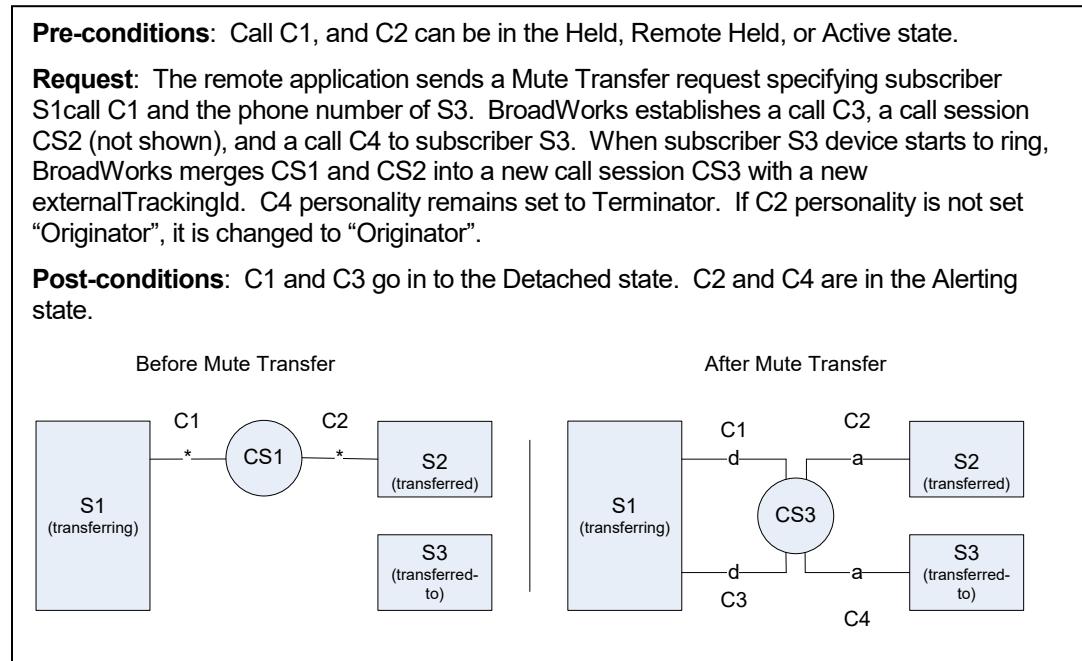


Figure 53 Mute Transfer State Changes

5.2.7.2 Request

A remote application can issue this request by sending a `MuteTransferRequest`. A `MuteTransferRequest` contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID associated with the call to transfer.
<code>callId</code>	CallId	Yes	The call ID of the call to transfer.
<code>address</code>	Address	Yes	The destination phone number for the transfer.

5.2.7.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>callStartInfo</code>	<code>callStartInfo</code>	No	Contains the call ID and external Tracking ID of the new call. It is only present when the <code>statusCode</code> is set to "201".

5.2.7.4 Associated Events

When a Mute Transfer request is accepted, the following events are generated:

- A *Call Originated* event against subscriber S1 for call C3.
- A *Call Received* event against subscriber S3 for call C4.
- A *Call Redirected* event against subscriber S1 for call C1 and C3.
- A *Call Transferred* event against the transferred subscriber S2 for call C2.
- A *Call Transferred* event against the transferred-to subscriber S3 for call C4.
- A *Hook Status* event against subscriber S1.

If the called destination is not available (for example, busy) then the following events are generated:

- A *Call Originated* event against subscriber S1 for call C3.
- A *Call Released* event against subscriber S1 for call C3 with the *releaseCause* properly set (for example busy).
- A *Call Released* event against subscriber S3 for call C4 with the *releaseCause* properly set (for example busy).

Note the following about subscriber S1 and S3:

- If S1 and S3 were not Cisco BroadWorks subscribers (that is, are just Network Parties) then no event would be generated against them and their associated call (C1 and C4).
- If S1 (or S3) was a Route Point then a *Route Point Call Updated* event would be generated in place of the *Call Transferred* event.
- If S1 (or S3) was an ACD the *ACD Call Updated* event would be generated in place of the *Call Transferred* event.

5.2.7.4.1 Impact on External Tracking ID

The mute transfer request is identical to transfer with consultation with respect to the call sessions and external tracking ID changes and resulting communication of those changes. Note that the consultation leg to S3 is generated with a unique external tracking ID. Then, upon completion of the mute transfer, the external tracking ID is updated for all parties involved in the transfer identically to a transfer with consultation action.

Specifically, the transferred calls, C2 for party S2 and C4 for party S3, are updated via a *Call Transferred* event for each monitored party. This means that the Call Transferred event will always carry the new external tracking ID for the call associated to the transferring party. The call ID in the *Call Transferred* event remains the same. The external tracking ID, the remote party, and the personality are updated for this scenario.

The transferring party's calls, C1 and C3 for party S1 are updated via a *Call Redirected* event with redirecting information for both calls being transferred together. The external tracking ID is updated in each call in the *Call Redirected* event. The call state is updated to *Detached* state for each call. A call redirect is included with a redirection reason of transfer for each call. The Call ID for calls C1 and C3, respectively, remains unchanged.



The Cisco BroadWorks CTI interface does not allow the Route Point to perform a mute transfer request so the Route Point can never be a transferring party. However, the Route point may be a transferred party. In this case, the Route Point is updated with a *Route Point Call Updated* event indicating that the Route Point queue entry has been transferred and the updated external tracking ID.

5.2.7.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	101001	State for call <callId> is not valid.
400	101003	Call <callId> is an Emergency call.
400	110540	The call <callId> is a Push-To-Talk call.
400	110010	Answer confirmation in progress for call <callId>.
400	110500	User is not assigned the Call Transfer service.
400	101000	Call <callId> has no remote call.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

For Address errors, see section [5.1.3.2 Address Error](#).

For Call ID errors, see section [5.1.3.3 CallId Errors](#).

5.2.8 Reconnect

5.2.8.1 Description

This request is used by a remote application to release all calls and retrieve a held call for a specific subscriber. A Reconnect can be performed only if the subscriber endpoint supports the protocol extensions to retrieve a call through a third-party call control command.

The call to reconnect must be in the *Held* state and the call must not be in a conference.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: The subscriber S1 is connected to subscriber S3 through call C3 and to subscriber S2 through call C1. Call C1 is the Held state, call C3 is in Held, Remote Held, and Alerting or Active state. Subscriber S1 endpoint supports the ability to retrieve a call through third party call control.

Request: The remote application sends a Reconnect request for subscriber S1 specifying that call C1 should be reconnected and call C3 should be released.

Post-conditions: The calls C1 and C2 are put back in the Active state or Remote Held/Held states as if a Talk was performed on C1. Call C3 is released unless it is in the Alerting state and has a “Terminator” personality, in which case C3 is maintained.

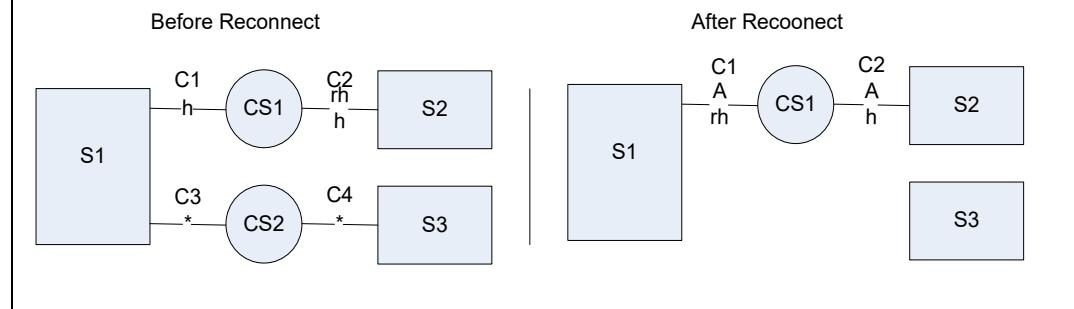


Figure 54 Reconnect State Changes

5.2.8.2 Request

A remote application can issue this request by sending a `ReconnectRequest`. A `ReconnectRequest` contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID associated with the call to reconnect.
<code>callId</code>	CallId	Yes	The call ID of the call to retrieve.

5.2.8.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.2.8.4 Associated Events

When the Reconnect request is accepted, the following events are generated:

- If call C1 goes into Active, a *Call Retrieved* event is generated against subscriber S1 for call C1 indicating that the call has been put back in the *Active* state. A *Call Retrieved* event is also generated against subscriber S2 for call C2.
- A *Call Released* event is generated against subscriber S1 for call C3. A *Call Released* event is also generated against subscriber S3 for call C4. A *Hook Status* event against S3.

Note that in the case where S2 and S3 were not Cisco BroadWorks subscribers (that is, are Network Parties) then no event would be generated against them and their associated call (C2 and C4).

5.2.8.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110581	Emergency call is present.
400	110610	Call <callId> is part of a conference.
400	100020	No endpoint for call <callId>.
400	101001	State for call <callId> is not valid.
400	100021	Endpoint state for call <callId> is not valid.
400	100006	Endpoint state is not valid.
400	110501	Consultation transfer in progress.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.2.9 Transfer to Voice Mail

5.2.9.1 Description

This request is used to transfer an existing call to a voice mail. This transfer is performed in a single step: that is, the client application doing the transfer does not have to place the existing call on hold before issuing the transfer.

If the voice mail destination is not specified, then the call is transferred to the subscriber's own Voice Mail.

The subscriber must have the Call Transfer service to use this request. There must also be a valid Cisco BroadWorks Voice Portal for the subscriber and/or target, or a valid Third-Party Voice Mail Support configuration for the target to use this request.

A call can be transferred when it is in the *Alerting*, *Held*, *Remote Held*, and *Active* state. When it is in the *Alerting* state, it must be have its personality set to "Terminator".

The following diagram provides information on the state and topology changes involved.

Pre-conditions: Call C1 can be in the Held, Remote Held, Alerting, or Active state.

Request: The remote application sends a Transfer to Voice Mail for subscriber S1 specifying call C1.

Post-conditions: Call C1 is released and goes into Detached. Call Session CS1 is maintained. Call C2 is in the Alerting state.

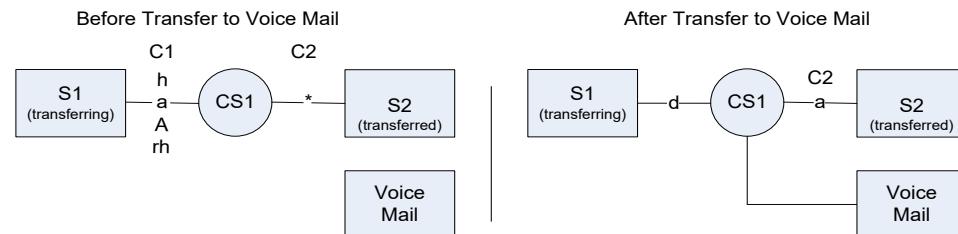


Figure 55 Transfer to Voice Mail

As shown in the diagram, because a Voice Mail is a Cisco BroadWorks virtual subscriber or a network party (in the case of Third-Party Voice Mail Support), no call is created between the call session and the Voice Mail. It is therefore not possible to receive events from this connection.

5.2.9.2 Request

A remote application can issue this request by sending a TransferToVoiceMailRequest. A TransferToVoiceMailRequest contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
subscriberId	SubscriberId	Yes	The subscriber ID associated with the call to transfer.
callId	CallId	Yes	The call ID of the call to transfer.
address	Address	No	The address of the subscriber who has the Voice Mail or Third-Party Voice Mail service. If not present, then the call is transferred to the subscriber's own Voice Mail.

5.2.9.3 Response

For information on the response, see section [5.1.2 Response](#).

5.2.9.4 Associated Events

When a Transfer to Voice Mail is accepted, the following events are generated:

- A *Call Redirected* event against subscriber S1 for call C1 indicating that the call has been redirected.
- A *Call Transferred* event against subscriber S2 for call C2 indicating that the call has been transferred.
- A *Hook Status* event against subscriber S1.



If the calling user is not available then the following events are generated:

- A Call Redirected event against subscriber S1 for call C1 indicating that the call has been redirected.
- A Call Transferred event against subscriber S2 for call C2 indicating that the call has been transferred with the releaseCause properly set.

NOTE: In the case where S2 is not a Cisco BroadWorks subscriber (that is, a network party), then no event is generated against its associated call (C2).

5.2.9.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	101003	Call <callId> is an Emergency call.
400	110500	User is not assigned the Call Transfer service.
400	101005	Diversions are inhibited for call <callId>.
400	101001	State for call <callId> is not valid.
400	110501	Consultation transfer in progress.
400	110000	Transfer target is not valid for the transfer to voice mail command.
400	100017	Blocked by maximum concurrent redirections policy.
400	101000	Call <callId> has no remote call.
400	110541	Call <callId> is an answered Push-To-Talk call.
400	100008	Remote party for call <callId> is a virtual subscriber.
400	100017	Blocked by maximum concurrent redirections policy.
401	100016	Blocked by Account/Authorization Codes service.
400	100018	Blocked by Intercept Group or Intercept User service.
400	100007	Blocked by Outgoing Calling Plan service.
400	100009	Blocked by Communication Barring service.
400	100010	Blocked by translations.
400	110507	Call <callId> with recall type <recallType> cannot be redirected prior to answer.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Address errors, see section 5.1.3.2 Address Error .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.2.10 Transmit DTMF

5.2.10.1 Description

This request is used to send a series of digits on behalf of a subscriber.



The following characters can be sent: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, #, A, B, C, and D. Note that a comma can be used to insert an additional delay. The tone generation attributes (such as, tone duration, pause duration, and inter-digit pause) are defined system wide and cannot be changed on a per request basis. The digits are sent either in-band or out-of-band based on system configuration and protocol negotiation.

The Play DTMF completes by itself. However, it can be stopped implicitly by a call state change (for example, if the call goes into the *Held* state or is released). The Play DTMF request can only be performed when the call is in the *Active* state. Only a single Play DTMF can be performed at a time.

This request does not involve any topology or call state change.

5.2.10.2 Request

A remote application can issue this request by sending a `TransmitDTMFRequest`. A `TransmitDTMFRequest` contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	<code>SubscriberId</code>	Yes	The subscriber ID associated with the call for which the DTMF will be transmitted.
<code>callId</code>	<code>CallId</code>	Yes	The call ID of the call to transmit the DTMF.
<code>playDTMF</code>	<code>String</code>	Yes	Specifies the list of DTMFs to send. It consists of the following set: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, #, A, B, C, D Note that a comma "," can be included in the parameter string to indicate a pause between characters. The length of the pause is defined by a system parameter.

5.2.10.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.2.10.4 Associated Events

There is no event associated to this request.

5.2.10.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	101000	Call <callId> has no remote call.
400	101001	State for call <callId> is not valid.
400	110540	The call <callId> is a Push-To-Talk call.
400	110810	DTMF transmission already in progress.
400	110811	DTMF string <dtmfString> is not valid. The string can contain only the following characters: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, #, A, B, C, D, ",".

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

5.2.11 Call Return

The Call Return request is used to initiate a call using the address of the last call missed or received by the subscriber. The user must have the Call Return service to use this request. A complete description of Call Return is provided in the *Cisco BroadWorks Service Guide [3]*.

When the request is received, Cisco BroadWorks performs basic validation and, if the request is accepted, it initiates a call towards the subscriber, assigns a call ID and an external tracking ID to the connection, and returns it in the response.

Similar to for the Dial request, Cisco BroadWorks alerts the subscriber's endpoint prompting the subscriber to go off-hook. In some cases, the endpoint may support protocol extensions to immediately go off-hook without requiring a subscriber action.

Once the subscriber endpoint has gone off-hook, two cases can occur:

- Case 1: The system is configured to dial directly the destination of the last received call. In this case, a new call session is created between the subscriber and the remote party.
- Case 2: The system is configured to work in a two-level mode, which provides announcements to guide the subscriber. In this case, the subscriber interacts with the system to confirm the destination of the last received call. If the confirmation is successful, a new call session is created between the subscriber and the remote party.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: Subscriber S1 and subscriber S2 are in the *Idle* state. Subscriber S2 was the last subscriber to call subscriber S1.

Request: The remote application sends a Call Return request for subscriber S1. When BroadWorks receives the request, it first validates it and, if the request is valid, create the call C1 and Call Session S1 and return the callId and externalTrackingId to the remote application. Subscriber S1 goes off-hook and a call C2 is established to subscriber S2.

Post-conditions: Call C1 and C2 are in the Alerting state.

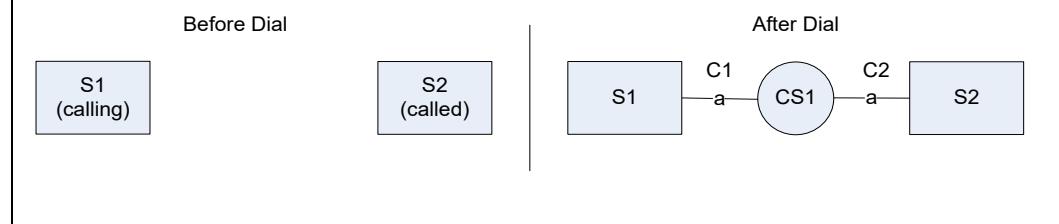


Figure 56 Call Return

5.2.11.1 Request

A remote application can issue this request by sending a CallReturnRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber who returns the call.



Parameter	Parameter Type	Required	Description
<i>location</i>	ClickToDialLocationEnum	No	When present, specifies the location of the calling user to ring for the call. When not present, locations are alerted per the calling user's profile on Cisco BroadWorks.
<i>locationAddress</i>	NonEmptyToken	No	<p>Specific address of location to ring. This element should only be present if the “location” element is present and set to “BroadWorksAnywhere”, “SharedCallAppearance”, or “Mobility”.</p> <p>When the <i>location</i> element is set to “BroadWorksAnywhere”, the locationAddress is optional and contains a directory number. The directory number may be in E.164 format or it may just contain digits. When not present, all BroadWorks Anywhere locations are alerted. The following are valid directory numbers for the locationAddress element:</p> <ul style="list-style-type: none"> ▪ 9725551000 ▪ +19725551000 <p>When the <i>location</i> element is set to “SharedCallAppearance”, the locationAddress is optional and contains a Cisco BroadWorks address of record. The address of record can be user@domain for SIP locations or user for MGCP locations. When not present, all Shared Call Appearance locations are alerted. The following are valid address of records for the locationAddress element:</p> <ul style="list-style-type: none"> ▪ 9725551000@broadsoft.com ▪ 9725551000 <p>When the “location” element is set to “Mobility”, the locationAddress is optional and contains a directory number. The directory number may be in E.164 format or it may just contain digits. When not present, all BroadWorks Mobility locations are alerted. The following are valid directory numbers for the locationAddress element:</p> <ul style="list-style-type: none"> ▪ 9725551000 ▪ +19725551000

5.2.11.2 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>callStartInfo</i>	CallStartInfo	No	Contains the call ID and external tracking ID of the new call. It is only present when the statusCode is set to “201”.

5.2.11.3 Associated Events

The following time sequence diagram illustrates the event produced when a Call Return is accepted for a system configured to work in two-level mode.

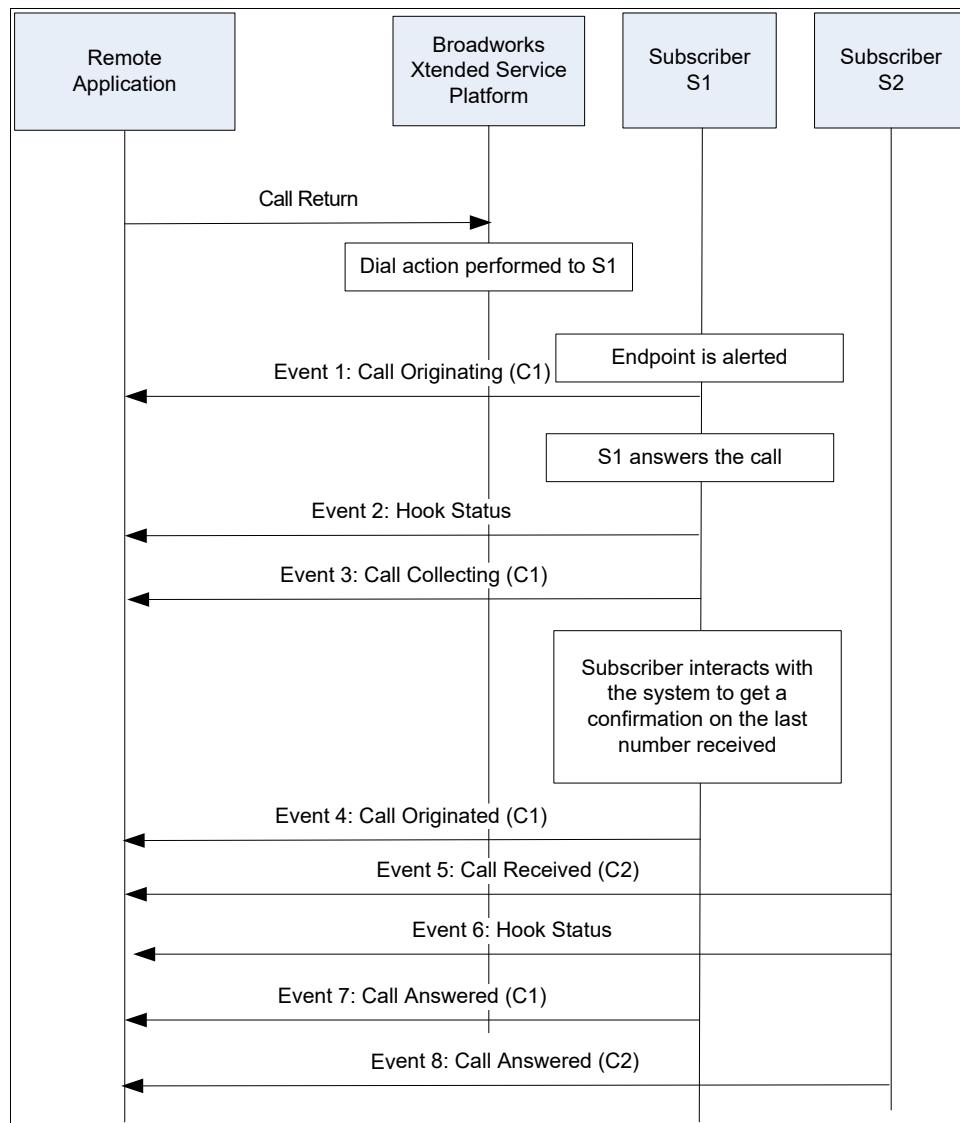


Figure 57 Call Return Events

A few additional remarks concerning this diagram:

- If the system is not configured to work in two-level mode then Event 2 and the subscriber interaction in the sequence diagram would not take place. All the other events remain.
- If the subscriber S1 is not available (for example, busy, device not registered, and so on.) then a *Call Released* event is sent after the *Call Originating* event with the appropriate releaseCause.



- If the called subscriber is invalid, then a *Call Originated* event after Event 1. A *Call Releasing* event is then generated with the releaseCause set to “User Not Found” as the subscriber is connected to a media treatment. A *Call Released* event is generated once the treatment stops playing or if the subscriber hangs up. The event also contains the same *releaseCause* value.
- If the call return fails (for example, denied by the system) than a *Call Releasing* and *Call Released* with the proper *releaseCause* are sent after Event 3.

Note that in the case where S1 was not Cisco BroadWorks subscriber (that is, are just Network Parties) then no event would be generated against S2 and its associated call (C2).

5.2.11.4 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111000	Invalid location value <location>.
400	111001	Cannot find location <locationAddress>.
400	111002	The locationAddress not allowed for location <location>.
400	111003	User is not assigned the <service name> service or there is no location assigned to this service.
400	111004	User does not have a primary endpoint.
400	110820	Blocked by Remote Office.
400	110560	User is not assigned the Call Return service.
400	100006	Endpoint state is not valid.
400	110581	Emergency call is present.
400	110582	Blocked by prepaid service.
400	110580	User has no charge address.
400	100019	Blocked by maximum simultaneous calls policy.
400	110011	Answer confirmation in progress.
400	110101	Invalid DNIS address.
400	110646	Not allowed to have more than one call outside the conference.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.2.12 Customer Originated Trace

5.2.12.1 Description

This request is used to trace a call. The trace is done by generating an SNMP alarm with the caller ID information. The subscriber must have the Customer Originated Trace (COT) service assigned to use this request successfully.

If the request specifies a call ID, then the trace is made for the corresponding call. This can only be done if the call is in the *Active*, *Held*, or *Remote Held* state with Terminator personality and can only be performed once per call. If the request does not specify a call ID, then the trace is made for the subscriber's most recently received/missed call.



Multiple invocation of COT on the same call will result in only one single trace being generated.

The execution of this request does not create session or a call.

5.2.12.2 Request

A remote application can issue this request by sending a CustomerOriginatedTraceRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID associated with the customer originated trace.
<code>callId</code>	CallId	No	The call ID for the call for which the Customer Originated Trace is requested. If not present, then the trace is made for the subscriber's most recently received/missed call.

5.2.12.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.2.12.4 Associated Events

There are no associated events for Customer Originated Trace request.

5.2.12.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110062	User is not assigned the Customer Originated Trace service.
400	110061	No valid call found to perform Customer Originated Trace.
400	110060	Customer Originated Trace already issued for call <callId>.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.2.13 Directed Call Pickup

The Directed Call Pickup request is used to pick up (answer) a call directed to another subscriber in the same customer group (or enterprise, if the group is part of an enterprise).

The following terms are introduced:

- DPU subscriber: The Direct Call Pickup (DPU) subscriber is the subscriber that is picking up the call.
- Picked up subscriber: The subscriber who is receiving the call to be picked up.
- The other party: The party that is dialing the picked up subscriber before the call is answered.

The DPU subscriber must have the Direct Call Pickup service to use this request. More details on the Directed Call Pickup are provided in the *Cisco BroadWorks Service Guide [3]*.

When the request is received, Cisco BroadWorks performs basic validation and, if the request is accepted, it initiates a call towards the DPU subscriber, assigns a call ID and an external tracking ID to the connection, and returns it in the response.

Similar to the Dial request, Cisco BroadWorks alerts the DPU subscriber's endpoint prompting the subscriber to go off-hook. In some cases, the endpoint may support protocol extensions to immediately go off-hook without requiring a subscriber action.

Once the DPU subscriber's endpoint has gone off-hook, then the call directed to the picked up subscriber is answered. A new call session is created between the DPU subscriber and the other party.

The following diagram provides information on the state and topology changes involved.

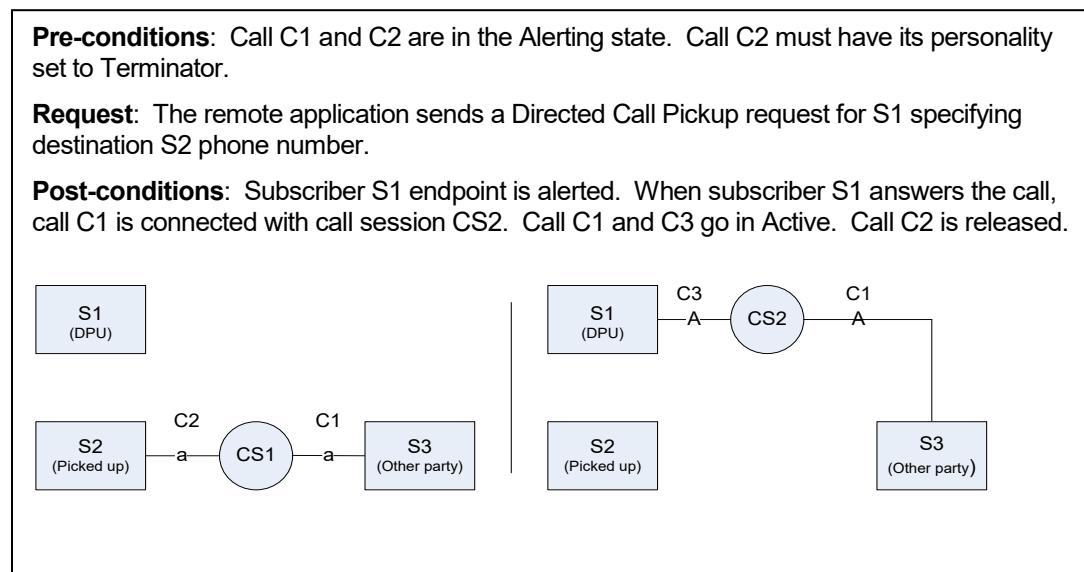


Figure 58 Directed Call Pickup

5.2.13.1 Request

The request contains the parameters in the following table.

A remote application can issue this request by sending a `DirectedCallPickupRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID performing the call pickup.
<code>address</code>	Address	Yes	The address of the subscriber to pick up against.
<code>location (AS only)</code>	ClickToDialLocationEnum	No	When present, specifies the location of the calling user to ring for the call. When not present, locations are alerted per the calling user's profile on Cisco BroadWorks.

Parameter	Parameter Type	Required	Description
<i>locationAddress</i> (AS only)	NonEmptyToken	No	<p>Specific address of location to ring. This element should only be present if the <i>location</i> element is present and set to "BroadWorksAnywhere", "SharedCallAppearance", or "Mobility".</p> <p>When the <i>location</i> element is set to "BroadWorksAnywhere", the locationAddress is optional and contains a directory number. The directory number may be in E.164 format or it may just contain digits. When not present, all BroadWorks Anywhere locations are alerted. The following are valid directory numbers for the locationAddress element:</p> <ul style="list-style-type: none"> ▪ 9725551000 ▪ +19725551000 <p>When the <i>location</i> element is set to "SharedCallAppearance", the locationAddress is optional and contains a Cisco BroadWorks address of record. The address of record can be user@domain for SIP locations or user for MGCP locations. When not present, all Shared Call Appearance locations are alerted. The following are valid address of records for the locationAddress element:</p> <ul style="list-style-type: none"> ▪ 9725551000@broadsoft.com ▪ 9725551000 <p>When the "location" element is set to "Mobility", the locationAddress is optional and contains a directory number. The directory number may be in E.164 format or it may just contain digits. When not present, all BroadWorks Mobility locations are alerted. The following are valid directory numbers for the locationAddress element:</p> <ul style="list-style-type: none"> ▪ 9725551000 ▪ +19725551000

5.2.13.2 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>callStartInfo</i>	CallStartInfo	No	Contains the call ID and external tracking ID of the new call. It is only present when the statusCode is set to "201".

5.2.13.3 Associated Events

The following time sequence diagram illustrates the event produced when a Directed Call Pickup request is accepted.

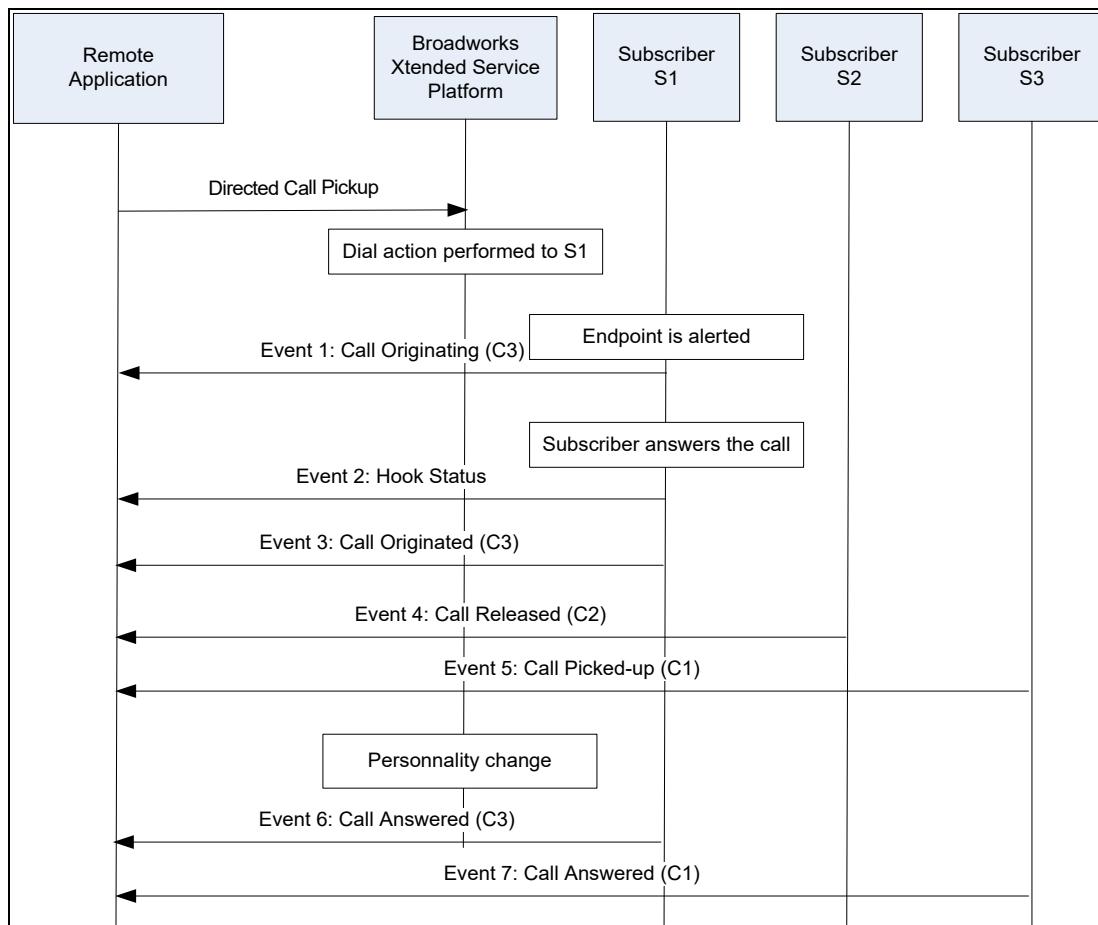


Figure 59 Directed Call Pickup Events

The following can be observed about these events:

- Event 5: The *Call Picked-up* event reports the new external tracking ID for call C1.
- Event 6: The *Call Answered* event reports the change in personality from Originator to Terminator for call C3.

If the Directed Call Pickup with Barge-in (DPUBI) subscriber is not available (for example, busy, device not registered, and so on.) then a *Call Released* event is sent after the *Call Originating* event with the appropriate releaseCause.

If the target subscriber is invalid, then a *Call Originating* event is first sent followed by a *Call Originated* event. A *Call Releasing* event is generated with the releaseCause set to "User Not Found" if the subscriber is connected to a media treatment. A *Call Released* event is generated once the treatment stops playing or if the subscriber hangs up. The event also contains the same releaseCause value.

Note that in the case where S3 was not Cisco BroadWorks subscriber (that is, Network Party) then no event would be generated against its associated call C1.



Note the following about subscriber S3:

- If S3 was not Cisco BroadWorks subscribers (that is, are just Network Party) then no event would be generated its associated call C1.
- If S3 was a Route Point then a *Route Point Call Updated* event would be generated in place of the *Call Picked-up* event.
- If S3 was an ACD the *ACD Call Updated* event would be generated in place of the *Call Picked-up* event.

5.2.13.3.1 Impact on External Tracking ID

The external tracking ID changes on the Directed Call Pickup request for the picked up party, S2.

This change is necessary as the Directed Pickup (DPU) party, S1, has a call for the directed call pickup with its own unique external tracking ID, and the picked up party, S2, has a separate unique external tracking ID for the call it received from party S3. The act of the pickup connects the calls of the DPU party, S1, and the other party, S3, where each party has a different external tracking ID. As a result of this action, the external tracking ID for the other party call is changed to the DPU party's external tracking ID.

The other party's call, C1 for party S3 is updated via a *Call Picked-up* event. This means that the *Call Picked-up* event will always carry the updated external tracking ID for the call associated to the DPU party, S1. The call ID in the *Call Picked-up* event remains the same and the remote party is updated to party S1.

The DPU party's call, C3 for party S1, is updated via a Call Updated event. The Call ID for call C3 remains the same. The external tracking ID remains unchanged. However, the personality is updated to terminator and the remote party is updated to party S3.

The *Call Picked-up* event will always indicate an external tracking ID change for the S3 party call.

5.2.13.4 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111000	Invalid location value <location>.
400	111001	Cannot find location <locationAddress>.
400	111002	The <i>locationAddress</i> not allowed for location <location>.
400	111003	User is not assigned the <service name> service or there is no location assigned to this service.
400	111004	User does not have a primary endpoint.
400	110820	Blocked by Remote Office.
400	110551	User is not assigned the Directed Call Pickup service.
400	100006	Endpoint state is not valid.
400	110581	Emergency call is present.
400	110582	Blocked by prepaid service.
400	110580	User has no charge address.
400	110646	Not allowed to have more than one call outside the conference.

statusCode	errorCode	Summary
400	100019	Blocked by maximum simultaneous calls policy.
400	110011	Answer confirmation in progress.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Address errors, see section 5.1.3.2 Address Error .		
For Address Destination errors, see 5.1.3.4 Address Destination Errors .		

5.2.14 Directed Call Pickup with Barge-in

The Directed Call Pickup with Barge-in (DPUBI) request is used to pick up (answer) a call directed to another subscriber in the same customer group (or enterprise, if the group is part of an enterprise), or barge in on the call if the call was already answered. When a barge-in occurs, a three-way call is established between the parties.

The following terms are introduced:

- DPUBI subscriber: The DPUBI subscriber is the subscriber that is picking up the call or barging-in on the call.
- Picked up subscriber: The subscriber who is receiving the call to be picked up.
- The other party: The party that is dialing the picked up subscriber before the call is answered.

The DPUBI subscriber must have the Direct Call Pickup service to use this request. More details on the DPUBI are provided in the *Cisco BroadWorks Service Guide* [3].

When the request is received, Cisco BroadWorks performs basic validation and if the request is accepted, then Cisco BroadWorks initiates a call towards the DPUBI subscriber, assigns a call ID and an external tracking ID to the connection, and returns it in the response.

Similar to the Dial request, Cisco BroadWorks alerts the DPUBI subscriber's endpoint prompting the subscriber to go off-hook. In some cases, the endpoint may support protocol extensions to immediately go off-hook without requiring a subscriber action.

Once the DPUBI subscriber endpoint has gone off-hook, then two cases are identified:

- Case 1: The picked up subscriber call is in the *Alerting* state with personality Terminator. In this case, the DPUBI subscriber and the other party are connected to one another and the picked up subscriber call is released. The functionality is identical to the Directed Call Pickup. For more information, see section [5.2.13 Directed Call Pickup](#).
- Case 2: The picked up subscriber's call has been answered (that is, it is in the *Held*, *Remote Held*, or *Active* state). In this case, a three-way call is established between the DPUBI, the picked up subscriber and the other party. The remaining of this section describes this case.

Once the DPUBI subscriber endpoint has gone off-hook, a barge-in occurs. If the subscriber's warning tone option is enabled, the picked up subscriber is given the barge-in warning tone. Note that the picked up user is not given the warning tone if he/she has put the call on hold. Once the warning tone has finished (or immediately if the subscriber's warning tone option is disabled), a three-way call is established with the DPUBI subscriber as the controller. The DPUBI subscriber has a call with the picked up subscriber and the other party. The picked up subscriber and the other party now have a call with the DPUBI subscriber instead of with each other.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: Call C1 and C2 can be in the Held, Remote Held, or Active state.

Request: The remote application sends a DPUBI on S1 with destination S2 phone number.

Post-conditions: Subscriber S1 endpoint is alerted. When subscriber S1 answers the call, a conference is created with S1 as the controller and S2 and S3 as participant. C1 and C2 states are set to the appropriate value. For more information, see state transition table.

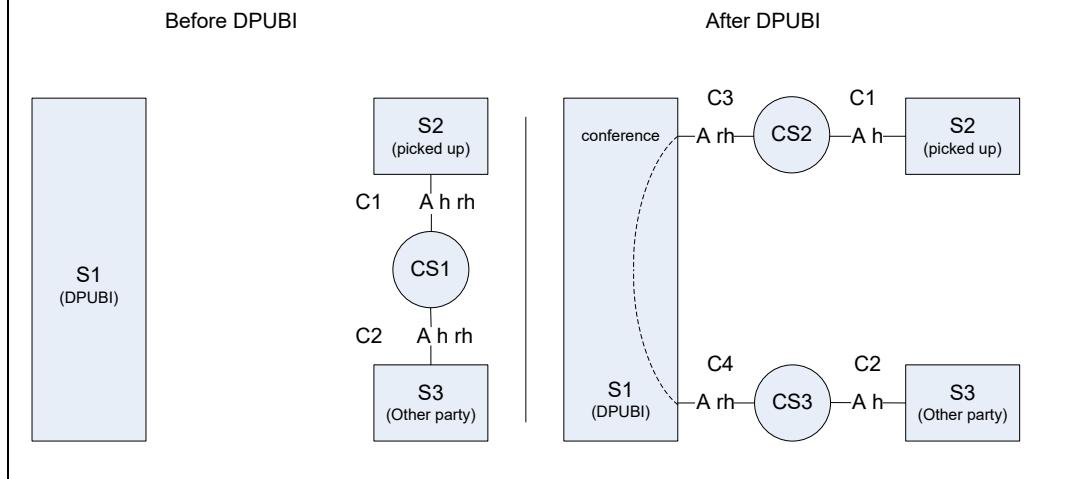


Figure 60 Directed Call Pickup with Barge-in

The following state transition table describes how call C1 and C2 state are affected by the DPUBI request.

Pre-DPUBI State			Post-DPUBI State		
C1 State	C2 State	C1 State	C3 State	C2 State	C4 State
Active	Active	Active	Active	Active	Active
Held	Remote Held	Held	Remote Held	Active	Active
Remote Held	Held	Active	Active	Remote Held	Held
Held	Held	Held	Remote Held	Held	Remote Held

Note that the request specifies the DPUBI subscriber (S1) and optionally the picked up subscriber address for call pickup (S2 phone number). If the Picked up subscriber address is not specified, then subscriber S1 is given stutter dial tones when answering the call so that he can enter the extensions to be picked up.

The Automatic Target Selection option can be enabled in which case the address for call pickup does not need to be specified. This can only work if S2 has only one subscriber extension is Active (on a call or ringing). If the target is ambiguous because multiple subscribers meet the condition, the DPUBI subscriber is given a stutter dial tone to enter the extension, as in the regular DPUBI service. If an invalid extension is entered (for example, an extension that does not exist in the group, too few digits, and so on), then the DPUBI subscriber is given reorder tone.

5.2.14.1 Release DPUBI Calls

The barge-in can be released at any time if the DPUBI subscriber hangs up the phone. The communication between the monitored subscriber and the other party is maintained. This is obtained by performing the equivalent of a Consultative Transfer on behalf of the Monitoring subscriber. For more information, see section [5.2.6 Consultative Transfer](#).

The following diagram provides information on the state and topology changes involved.

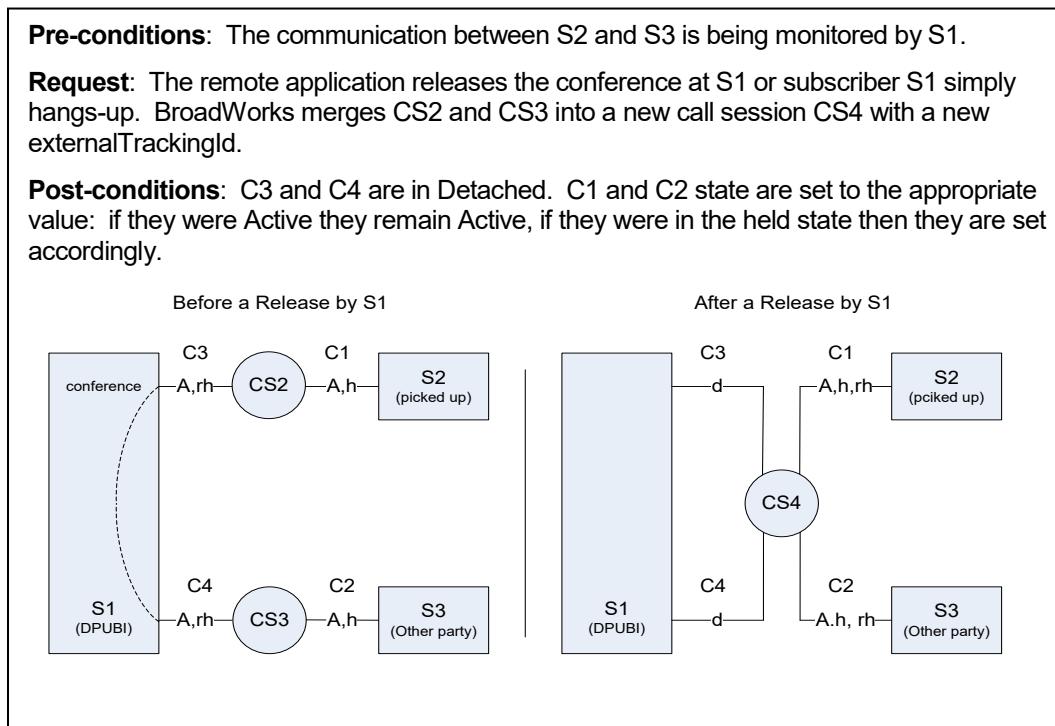


Figure 61 Directed Call Pickup with Barge-in State Changes

5.2.14.2 Request

The request contains the parameters in the following table.

A remote application can issue this request by sending a `DirectedCallPickupWithBargeIn` Request. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID performing the call barge-in.
<code>address</code>	Address	No	The address of the subscriber to pick up or barge in against. If not present, then automatic target selection is used if enabled or the subscriber is prompted to enter the target address upon answering the Dial attempt.
<code>location (AS only)</code>	ClickToDialLocationEnum	No	When present, specifies the location of the calling user to ring for the call. When not present, locations are alerted per the calling user's profile on Cisco BroadWorks.

Parameter	Parameter Type	Required	Description
<i>locationAddress</i> (AS only)	NonEmptyToken	No	<p>Specific address of location to ring. This element should only be present if the “location” element is present and set to “BroadWorksAnywhere”, “SharedCallAppearance”, or “Mobility”.</p> <p>When the “location” element is set to “BroadWorksAnywhere”, the locationAddress is optional and contains a directory number. The directory number may be in E.164 format or it may just contain digits. When not present, all BroadWorks Anywhere locations are alerted. The following are valid directory numbers for the locationAddress element:</p> <ul style="list-style-type: none"> ▪ 9725551000 ▪ +19725551000 <p>When the <i>location</i> element is set to “SharedCallAppearance”, the locationAddress is optional and contains a Cisco BroadWorks address of record. The address of record can be user@domain for SIP locations or user for MGCP locations. When not present, all Shared Call Appearance locations are alerted. The following are valid address of records for the <i>locationAddress</i> element:</p> <ul style="list-style-type: none"> ▪ 9725551000@broadsoft.com ▪ 9725551000 <p>When the <i>location</i> element is set to “Mobility”, the locationAddress is optional and contains a directory number. The directory number may be in E.164 format or it may just contain digits. When not present, all BroadWorks Mobility locations are alerted. The following are valid directory numbers for the locationAddress element:</p> <ul style="list-style-type: none"> ▪ 9725551000 ▪ +19725551000

5.2.14.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>callStartInfo</i>	CallStartInfo	No	Contains the call ID and external tracking ID of the new call (C3 in the diagram). It is only present when the statusCode is set to “201”.

5.2.14.4 Associated Events

The following time sequence diagram illustrates the event produced when a Directed Call Pickup with Barge-In request is accepted.

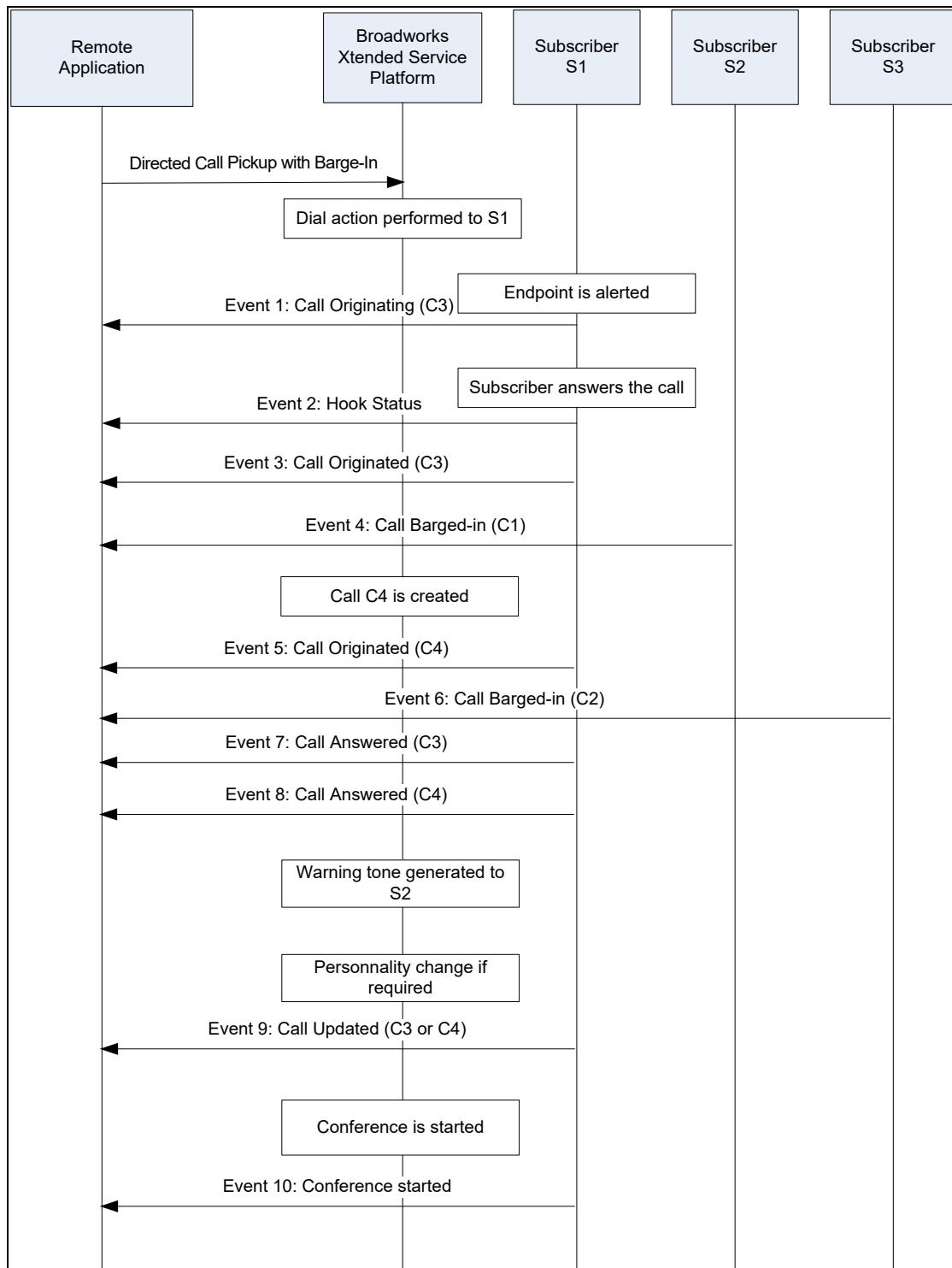


Figure 62 Events



The following can be observed about these events.

- Event 4: The Barged-in event reports the new external tracking ID for call C1 and C2. The answer time reported in the event is changed to the time the barge in occurred.
- Event 9: The Call Updated reports any change in personally for call C3 and C4. Both calls were created with personality Originator so one of the two personalities is changed here.

If the DPUBI subscriber is not available (for example, busy, device not registered, and so on.) then a *Call Released* event is sent after the *Call Originating* event with the appropriate releaseCause.

If the target subscriber is invalid, then a *Call Originating* event is first sent followed by a *Call Originated*. A *Call Releasing* event is generated with the releaseCause set to "User Not Found" if the subscriber is connected to a media treatment. A *Call Released* event is generated once the treatment stops playing or if the subscriber hangs up. The event also contains the same releaseCause value.

Note the following about subscriber S2 and S3:

- If S2 (or S3) was not a Cisco BroadWorks subscriber (that is, is just Network Party) then no event would be generated against its associated call C1 (or C2).
- If S2 (or S3) was a Route Point then a *Route Point Call Updated* event would be generated in place of the *Call Barged-in* event.
- If S2 (or S3) was an ACD the *ACD Call Updated* event would be generated in place of the *Call Barged-in* event.

5.2.14.4.1 Impact on External Tracking ID

The external tracking ID will change on the Directed Call Pickup action for the picked-up party, S2 and for the other party, S3.

These changes are necessary as the DPU party, S1, has a call for the directed call pickup with barge-in with its own unique external tracking ID, the picked up party, S2, has a separate unique external tracking ID for the call it received from party S3, and the other party S3, has the external tracking ID for the call it placed to party S2. The act of the pickup connects the calls of the DPU party, S1, and the picked up party, S2, where each party has a different external tracking ID. As a result of this action, the external tracking ID for the picked up party call is changed to the DPUBI party's external tracking ID. Additionally, the party S3's call C2 is connected to a new call C4 to the DPU party, S1. As a result of this action, the external tracking ID is updated for the C2 call for party S3 to the external tracking ID for call C4 on the DPUBI party, S1.

The other party's call, C2 for party S3 is updated via a *Call Barged-in* event. This means that the *Call Barged-in* event will always carry the updated external tracking ID for the call associated to the DPU party, C4 for party S1. The call ID in the *Call Barged-in* event remains the same for call C2 for party S3. The external tracking ID is updated to the external tracking ID for call C4 for party S1 and the remote party is updated to party S1.

The picked up party's call, C1 for party S2 is updated via a *Call Barged-in* event. This means that the *Call Barged-in* event will always carry the updated external tracking ID for the call associated to the DPU party, C3 for party S1. The call ID in the *Call Barged-In* event remains the same for call C1 for party S2. The external tracking ID is updated to the external tracking ID for call C3 for party S1 and the remote party is updated to party S1.

The external tracking ID for the DPU party's calls, C3 and C4 (for party S1) does not change from the creation of the respective calls.

The *Call Barged-in* event always indicates an external tracking ID change for the S2 party C1 call and the S3 party C2 call.



5.2.14.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111000	Invalid location value <location>.
400	111001	Cannot find location <locationAddress>.
400	111002	The <i>locationAddress</i> not allowed for location <location>.
400	111003	User is not assigned the <service name> service or there is no location assigned to this service.
400	111004	User does not have a primary endpoint.
400	110820	Blocked by Remote Office.
400	110552	User is not assigned the Directed Call Pickup with Barge-in service.
400	100006	Endpoint state is not valid.
400	110581	Emergency call is present.
400	110582	Blocked by prepaid service.
400	110580	User has no charge address.
400	110646	Not allowed to have more than one call outside the conference.
400	100019	Blocked by maximum simultaneous calls policy.
400	110011	Answer confirmation in progress.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

For Address errors, see section [5.1.3.2 Address Error](#).

For Address Destination errors, see section [5.1.3.4 Address Destination Errors](#).

5.2.15 Monitor Call

The Monitor Call request is used to silently monitor an existing call handled by an agent in the same customer group (or enterprise, if the group is part of an enterprise). When a monitor call occurs, a three-way call is established between the parties.

The following terms are introduced:

- Monitoring subscriber: The subscriber that is listening to the call.
- Monitored subscriber: The subscriber who is receiving the call to be monitored (that is, the agent)
- The other party: The party that is connected to the monitored subscriber.

The monitoring subscriber must have the Direct Call Pickup with Barge-in service and the Call Center Monitoring service to use this request.

When the request is received, Cisco BroadWorks performs basic validation and if the request is accepted, then Cisco BroadWorks initiates a call towards the monitoring subscriber, assigns a call ID and an external tracking ID to the connection, and returns it in the response.

As for any Dial request, Cisco BroadWorks alerts the monitoring subscriber's endpoint prompting the subscriber to go off-hook. In some cases, the endpoint may support protocol extensions to immediately go off-hook without requiring a subscriber action.

Once the subscriber endpoint has gone off-hook, then the call monitoring can start. If the subscriber's monitoring warning tone option is enabled, the monitored subscriber is given a warning tone. Note that the monitored subscriber is not given the warning tone if he/she has put the call on hold. Once the warning tone has finished (or immediately if the subscriber's warning tone option is disabled), a three-way call is established with the monitoring subscriber as the controller. The monitoring subscriber has a call with the picked up subscriber and another call with the other party. The picked up subscriber and the other party now have a call with the monitoring subscriber instead of with each other. However, the monitored subscriber and the other party cannot hear the monitoring subscriber.

Note that if the monitored subscriber has no call in the *Active*, *Held*, or *Remote Held* state, than the request is aborted and the monitoring subscriber is played a treatment.

The following diagram provides information on the state and topology changes involved.

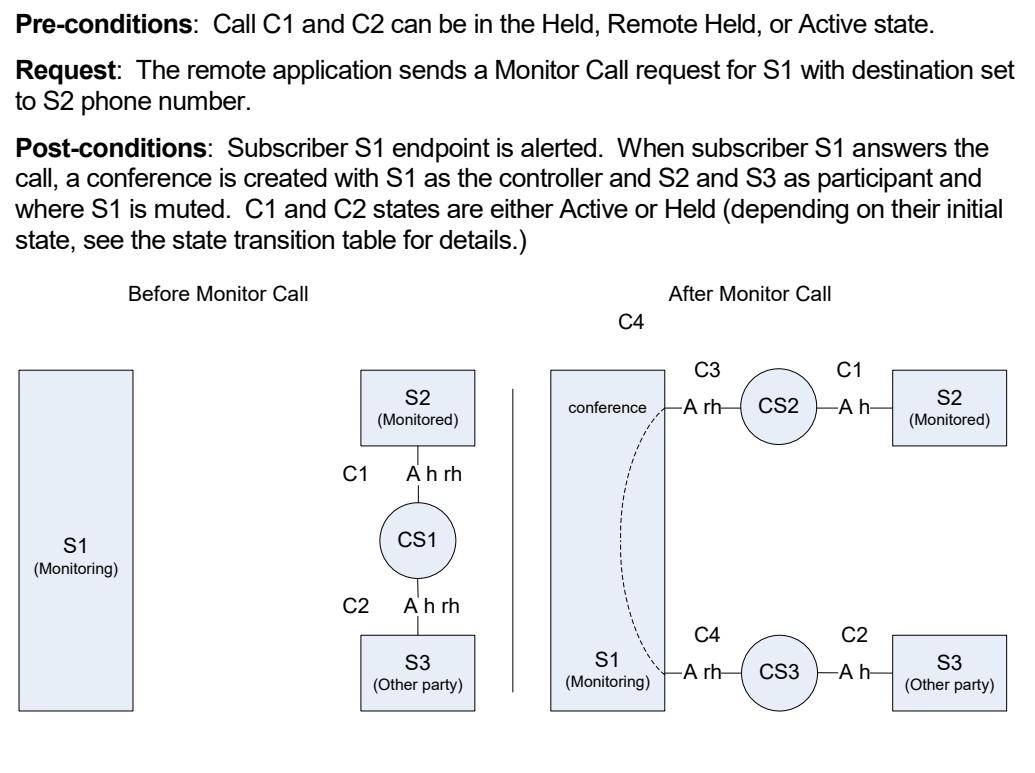


Figure 63 Monitor Call

The following state transition table describes how call C1 and C2 states are affected by the Monitor request.

Pre-Monitor State		Post-Monitor State			
C1 State	C2 State	C1 State	C3 State	C2 State	C4 State
Active	Active	Active	Active	Active	Active
Held	Remote Held	Held	Remote Held	Active	Active

Pre-Monitor State			Post-Monitor State		
C1 State	C2 State	C1 State	C3 State	C2 State	C4 State
Remote Held	Held	Active	Active	Remote Held	Held
Held	Held	Held	Remote Held	Held	Remote Held

5.2.15.1 Releasing Monitor Call

The Monitor call can be released at any time if the Monitoring subscriber hangs up the phone. The communication between the monitored subscriber and the other party is maintained. This is obtained by performing the equivalent of a Consultative Transfer on behalf of the Monitoring subscriber. The following diagram provides information on the state and topology changes involved.

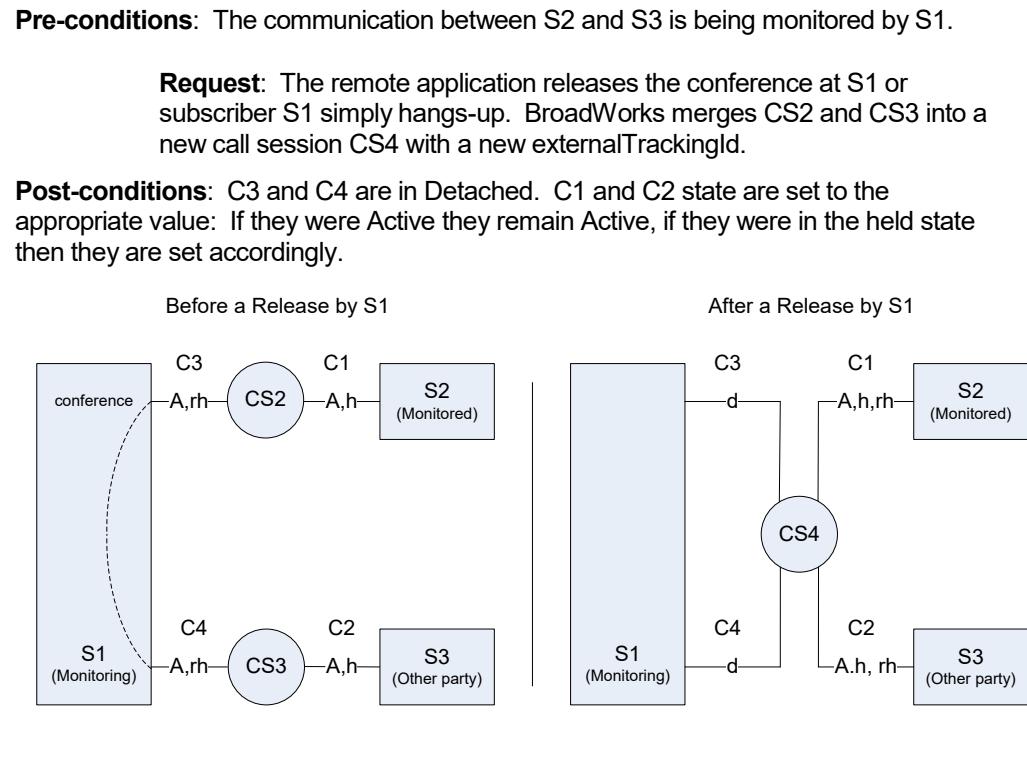


Figure 64 Monitor Call State Changes

5.2.15.2 Request

A remote application can issue this request by sending a MonitorCallRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID performing the monitor call.
<i>address</i>	Address	Yes	The address of the monitored subscriber.

Parameter	Parameter Type	Required	Description
<i>location</i>	ClickToDialLocationEnum	No	When present, specifies the location of the calling user to ring for the call. When not present, locations are alerted per the calling user's profile on Cisco BroadWorks.
<i>locationAddress</i>	NonEmptyToken	No	<p>Specific address of location to ring. This element should only be present if the "location" element is present and set to "BroadWorksAnywhere", "SharedCallAppearance", or "Mobility".</p> <p>When the <i>location</i> element is set to "BroadWorksAnywhere", the locationAddress is optional and contains a directory number. The directory number may be in E.164 format or it can just contain digits. When not present, all BroadWorks Anywhere locations are alerted. The following are valid directory numbers for the locationAddress element:</p> <ul style="list-style-type: none"> ▪ 9725551000 ▪ +19725551000 <p>When the <i>location</i> element is set to "SharedCallAppearance", the locationAddress is optional and contains a Cisco BroadWorks address of record. The address of record can be user@domain for SIP locations or user for MGCP locations. When not present, all Shared Call Appearance locations are alerted. The following are valid address of records for the locationAddress element:</p> <ul style="list-style-type: none"> ▪ 9725551000@broadsoft.com ▪ 9725551000 <p>When the <i>location</i> element is set to "Mobility", the locationAddress is optional and contains a directory number. The directory number may be in E.164 format or it may just contain digits. When not present, all BroadWorks Mobility locations are alerted. The following are valid directory numbers for the locationAddress element:</p> <ul style="list-style-type: none"> ▪ 9725551000 ▪ +19725551000

5.2.15.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>callStartInfo</i>	CallStartInfo	No	Contains the call ID and external tracking ID of the new call. It is only present when the statusCode is set to "201".

5.2.15.4 Associated Events

The following time sequence diagram illustrates the event produced when a Monitor Call is accepted.

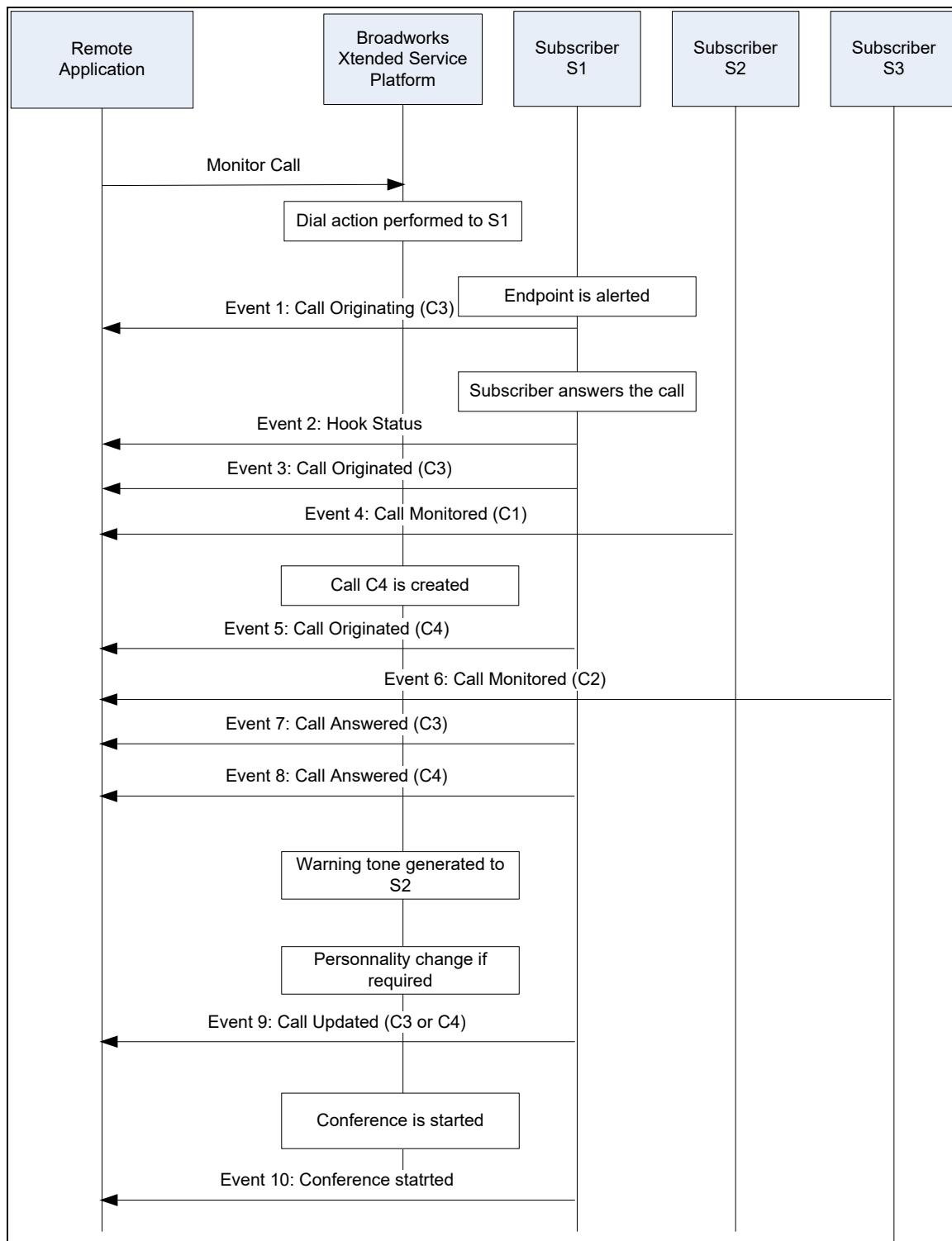


Figure 65 Monitor Call State Changes



The following can be observed about these events:

- Event 4: The Monitored event reports the new external tracking ID for call C1 and C2. The answer time reported in the event is changed to the time the monitoring started.
- Event 9: The Call Updated reports any change in personality for call C3 and C4. Both calls were created with personality Originator so one of the two personalities must be changed.

If the monitoring subscriber is not available (for example, busy, device not registered, and so on,) then a *Call Released* event is sent after the *Call Originating* event with the appropriate *releaseCause*.

If the monitoring call cannot be established, then a *Call Originating* event is first sent followed by a *Call Originated*. A *Call Releasing* event is generated with the *releaseCause* set to “User Not Found” if the subscriber is connected to a media treatment. A *Call Released* event is generated once the treatment stops playing or if the subscriber hangs up. The event also contains the same *releaseCause* value.

If subscriber S2 is involved with more than one call, Cisco BroadWorks cannot determine the call to be monitored. A *Call Releasing* event is generated after Event 2 to notify the subscriber of the request failure.

Note the following about subscriber S2 and S3:

- If S2 (or S3) was not a Cisco BroadWorks subscriber (that is, is a network party) then no event would be generated against its associated call C1 (or C2).
- If S2 (or S3) was a Route Point then a *Route Point Call Updated* event would be generated in place of the *Call Monitored* event.
- If S2 (or S3) was an ACD then an *ACD Call Updated* event would be generated in place of the *Call Monitored* event.

5.2.15.4.1 Impact on External Tracking ID

The external tracking ID will change on the Monitor Call action for the monitored party, S2, and for the other party, S3.

From a call topology point of view, the Monitor Call request is similar to a Directed Call Pickup with Barge-in action.

The external tracking ID changes are necessary as the monitoring party, S1, has a call for the monitor call with its own unique external tracking ID, the monitored party, S2, has a separate unique external tracking ID for the call it received from party S3, and the other party S3, has the external tracking ID for the call it placed to party S2. The act of the monitor call connects the calls of the monitoring party, S1, and the monitored party, S2, where each party has a different external tracking ID. As a result of this action, the external tracking ID for the monitored party call is changed to the monitoring party's external tracking ID. Additionally, the party S3's call C2 is connected to a new call C4 to the monitoring party, S1. As a result of this action, the external tracking ID is updated for the C2 call for party S3 to the external tracking ID for call C4 on the monitoring party, S1.

The other party's call, C2 for party S3 is updated via a *Call Monitored* event. This means that the *Call Monitored* event will always carry the updated external tracking ID for the call associated to the monitoring party, C4 for party S1. The call ID in the *Call Monitored* event remains the same for call C2 for party S3. The external tracking ID is updated to the external tracking ID for call C4 for party S1.



The monitored party's call, C1 for party S2 is updated via a *Call Monitored* event. This means that the *Call Monitored* event will always carry the updated external tracking ID for the call associated to the monitoring party, C3 for party S1. The call ID in the *Call Monitored* event remains the same for call C1 for party S2. The external tracking ID is updated to the external tracking ID for call C3 for party S1.

The external tracking ID for the monitoring party's calls, C3 and C4 for party S1 does not change from the creation of the respective calls.

The *Call Monitored* event will always indicate an external tracking ID change for the S2 party C1 call and the S3 party C2 call.

5.2.15.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111000	Invalid location value <location>.
400	111001	Cannot find location <locationAddress>.
400	111002	locationAddress not allowed for location <location>.
400	111003	User is not assigned the <service name> service or there is no location assigned to this service.
400	111004	User does not have a primary endpoint.
400	110820	Blocked by Remote Office.
400	110105	User is not a supervisor.
400	110552	User is not assigned the Directed Call Pickup with Barge-in service.
400	110760	User is not assigned the Call Center Monitoring service.
400	100006	Endpoint state is not valid.
400	110581	Emergency call is present.
400	110582	Blocked by prepaid service.
400	110580	User has no charge address.
400	110646	Not allowed to have more than one call outside the conference.
400	100019	Blocked by maximum simultaneous calls policy.
400	110011	Answer confirmation in progress.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

For Address errors, see section [5.1.3.2 Address Error](#).



5.2.16 Monitor Next Call (**AS only**)

The Monitor Next Call request is used to silently monitor the next call received by a subscriber in the same customer group (or enterprise, if the group is part of an enterprise).

The following terms are used:

- Monitoring subscriber: The subscriber that will be listening to the call.
- Monitored subscriber: The subscriber who is receiving the call to be monitored. This can either be an ACD, a Route Point or an agent.
- The other party: The party that is dialing the monitored subscriber before the call is answered.

The Monitoring subscriber must have the Direct Call Pickup with Barge-In and Call Center Monitoring service to use this request. Also, note that a monitored subscriber can be monitored by one subscriber at a time.

Three main cases are identified as follows:

- Monitored subscriber is a Route Point
- Monitored subscriber is an ACD
- Monitored subscriber is an agent

The following subsections provide more details for each case.

5.2.16.1 Monitor Next Call on a Route Point

The execution of the Monitor Next Call on a Route Point is performed in two steps:

When the request is received, Cisco BroadWorks performs basic validation and if the request is accepted, then Cisco BroadWorks initiates a call towards the monitoring subscriber, assigns a call ID and an external tracking ID to the connection, and returns it in the response.

Similar to the Dial request, Cisco BroadWorks alerts the monitoring subscriber's endpoint prompting the subscriber to go off-hook. In some cases, the endpoint may support protocol extensions to immediately go off-hook without requiring a subscriber action.

Once the subscriber endpoint has gone off-hook, then the call goes in the *Active* state and the subscriber receives silence until connected to the next call received by the Route Point.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: Subscriber S1 is the Idle state.

Request: The remote application sends a Monitor Next Call request for S1 with Route Point address.

Post-conditions: Subscriber S1 endpoint is alerted. When subscriber S1 answers the call, subscriber S1 receives silence. Call C1 is in the Active State.

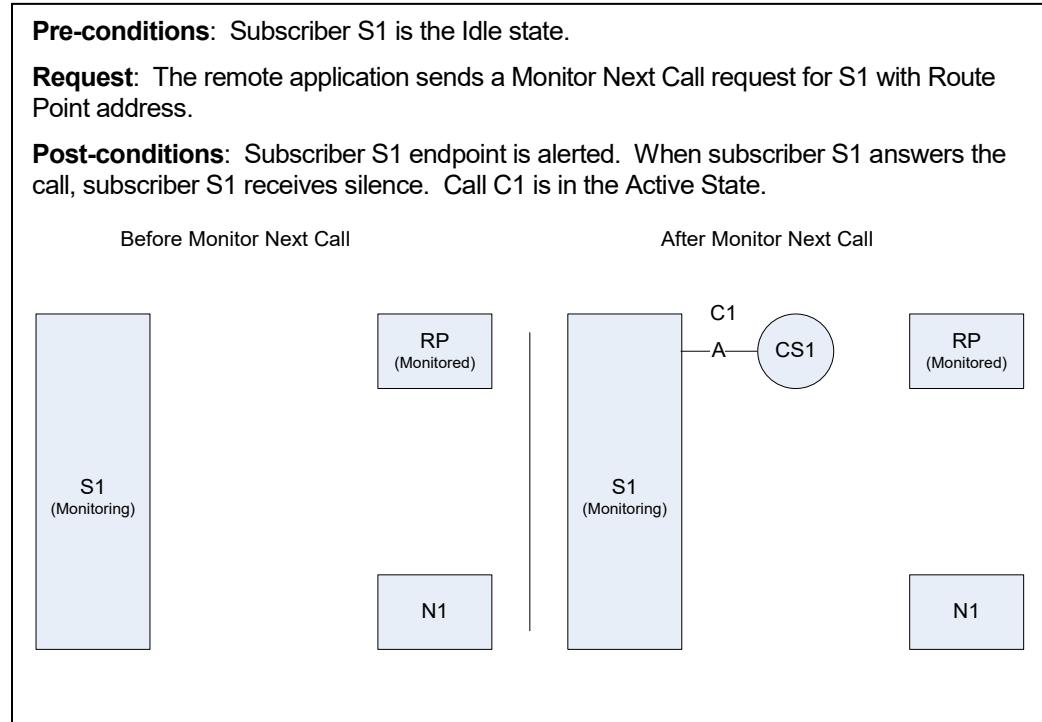


Figure 66 Monitor Next Call

When the Route Point receives an incoming call from a network party (N1), a three-way call is established with the monitoring subscriber as the controller. The monitoring subscriber has a call with the Route Point and another call with the network party (N1). The Route Point and the network party now have a call with the monitoring subscriber instead of with each other. However, the network party cannot hear the monitoring subscriber.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: A Monitor Next Call has been initiated on behalf of S1 identifying RP as the Monitored subscriber. The network party N1 dials RP phone number and a call session CS2 is established. Call C2 is in Alerting.

Request: None (the call C2 automatically goes into the Active state).

Post-conditions: A conference is created with S1 as the controller and RP and N1 as participants. All calls are in Active. The subscriber S1 is muted so that N1 does not receive any media from S1.

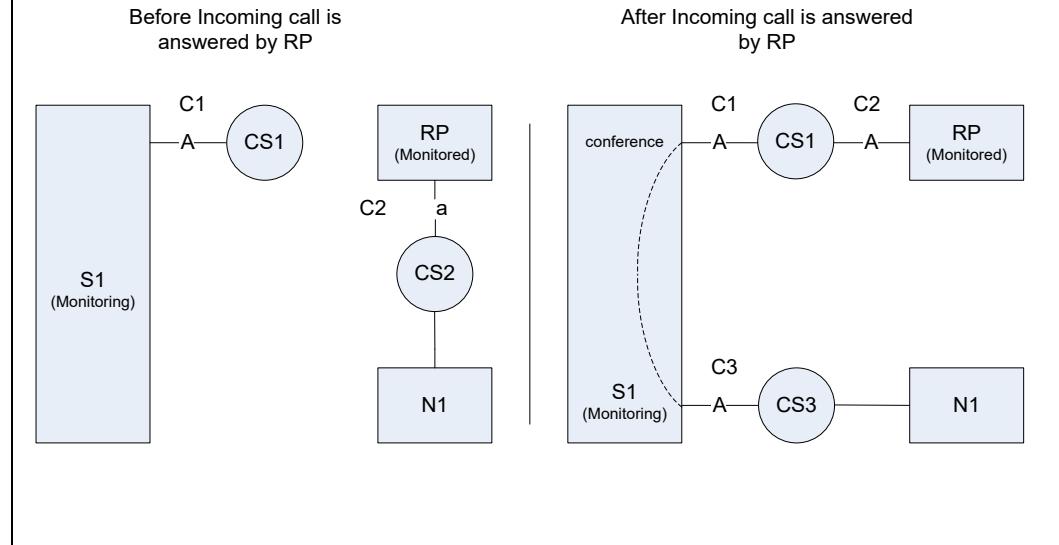


Figure 67 Monitor Next Call State Changes

5.2.16.1.1 Release Monitor Next Call

The Monitor Next Call can be released at any time if the Monitoring subscriber hangs up the phone. The communication between the Route Point and the other party is maintained. This is obtained by performing the equivalent of a Consultative Transfer on behalf of the Monitoring subscriber.

The following diagram provides information on the state and topology changes involved.

- Pre-conditions:** The communication between N1 and RP is being monitored by S1.
- Request:** The remote application releases the conference at S1 or subscriber S1 simply hangs-up. BroadWorks merges CS1 and CS3 into a new call session CS4 with a new externalTrackingId.
- Post-conditions:** C1 and C3 are in Detached.

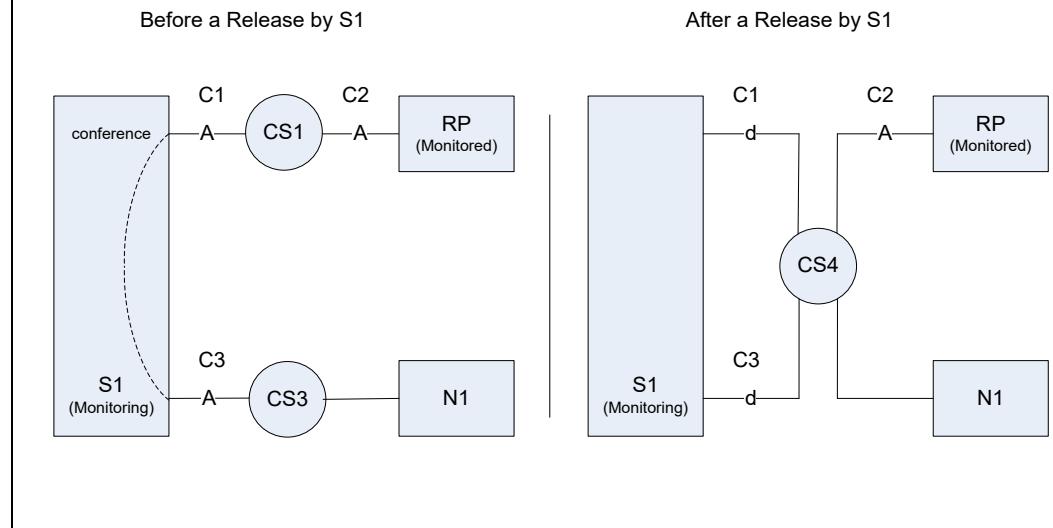


Figure 68 Releasing Monitor Next Call

5.2.16.1.2 Associated Events

The following time sequence diagram illustrates the event produced when a Monitor Next Call request is accepted.

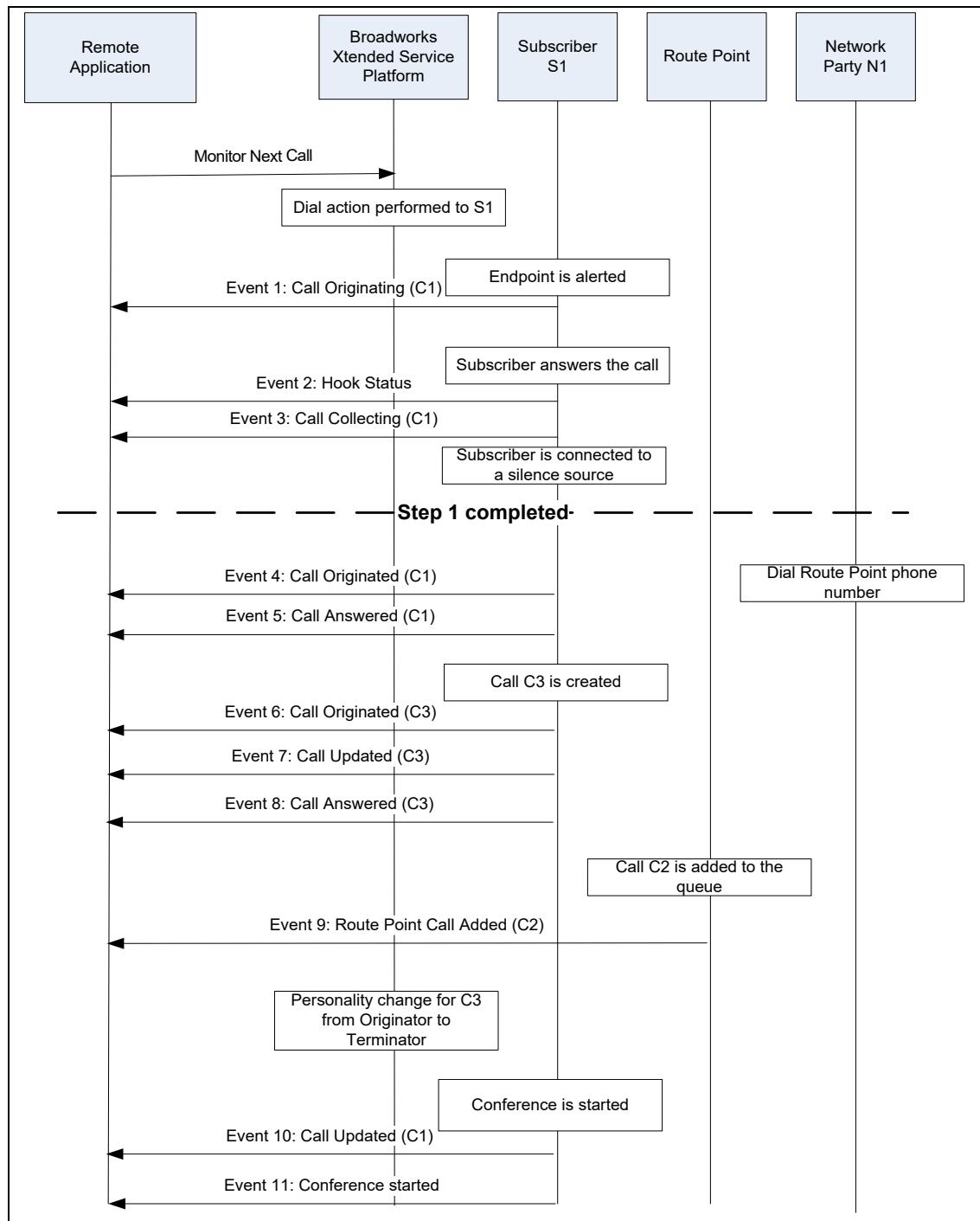


Figure 69 Releasing Monitor Next Call State Changes



During the first step, if the monitoring subscriber is not available (for example, busy, device not registered, and so on) then a *Call Released* event is sent after the *Call Originating* event with the appropriate releaseCause. Note that the *Call Updated* (Event 7) reports the change in personality for call C3, which was initially created with the Originator personality.

If the monitoring call cannot be established (for example, the address of the monitored subscriber is invalid), then a *Call Originating* event is first sent followed by a *Call Originated* event. A *Call Releasing* event is generated with the releaseCause set to "User Not Found" if the subscriber is connected to a media treatment. A *Call Released* event is generated once the treatment stops playing or if the subscriber hangs up. The event also contains the same *releaseCause* value.

5.2.16.2 Monitor Next Call on an ACD

The Monitor Next Call on an ACD performs exactly the same way as for a Route Point. The only differences are:

- The *Route Point Call Added* event is replaced by an *ACD Call Added* event. This event contains the initial external tracking ID (CS2 in the diagram).
- An ACD Call Updated event is sent after the ACD Call Added to report the new external tracking ID of the call (CS1 in the diagram).

5.2.16.3 Monitor Next Call on an Agent

The execution of the Monitor Next Call on an agent is also performed in two steps.

The first step is the same as for a Route Point and an ACD where the monitoring subscriber is connected to Cisco BroadWorks and receives silence until connected to the next ACD or Route Point call answered by the agent.

The second step is a variation on the ACD and Route Point cases. When the agent receives an incoming call from a Route Point or an ACD and answers it, then the call monitoring can start. If the monitoring subscriber warning tone option is enabled, the agent is given a warning tone. Note that the agent is not given the warning tone if he/she has put the call on hold. Once the warning tone has finished (or immediately if the agent's warning tone option is disabled), a three-way call is established with the monitoring subscriber as the controller. The monitoring subscriber has a call with the agent and another call with the other party. The agent and the other party now have a call with the monitoring subscriber instead of with each other. However, the agent and the other party cannot hear the monitoring subscriber.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: A Monitor Next Call has been initiated on behalf of S1 identifying A1 as the Monitored subscriber. The network party N1 is in a queue at RP and call C3 is offered to the agent.

Request: The agent answers call C3.

Post-conditions: A conference is created with S1 as the controller and A1 and N1 as participants. All calls are in the Active state. The subscriber S1 is muted so that A1 and N1 do not receive any media from S1.

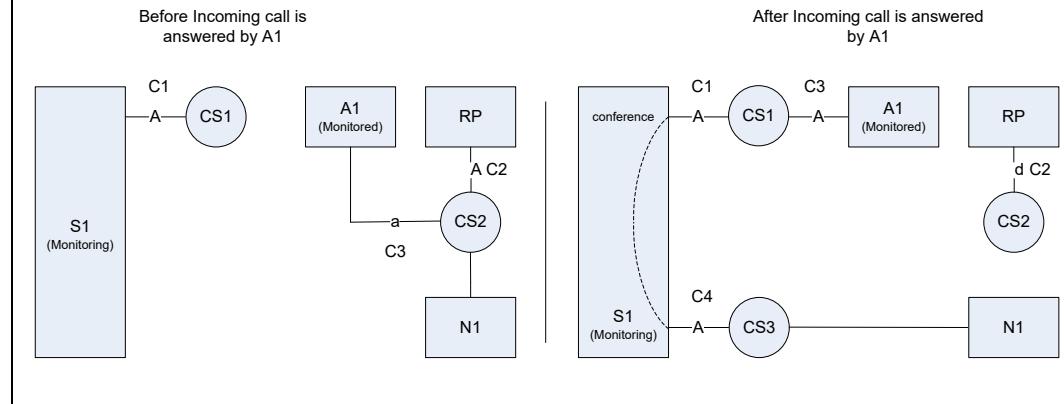


Figure 70 Monitor Next Call on Agent

5.2.16.3.1 Associated Events

The following time sequence diagram illustrates the event produced during step 2 of a Monitor Next Call on an Agent.

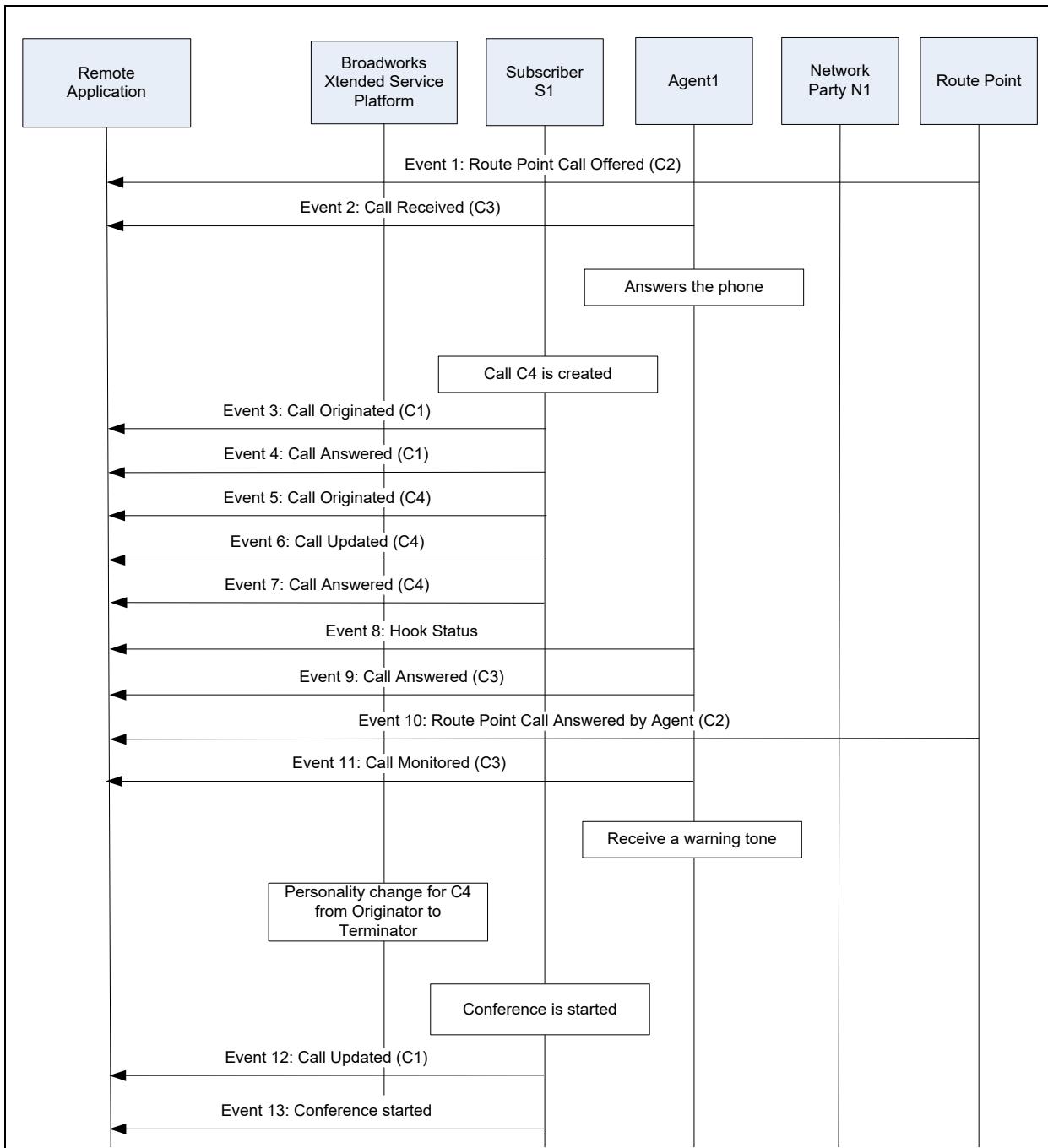


Figure 71 Monitor Next Call onAgent Events

5.2.16.4 Impact on External Tracking ID

The external tracking ID will change on the Monitor Next Call action for the monitored party (Route Point, ACD, or agent).



The external tracking ID changes are necessary as the monitoring party, S1, has a call for the monitor next call with its own unique external tracking ID, the monitored party, Route Point, or A1, has a separate unique external tracking ID for the call it received from party N1. The act of the monitor next call connects the calls of the monitoring party, S1, and the monitored party, Route Point, or A1, where each party has a different external tracking ID. As a result of this action, the external tracking ID for the monitored party call is changed to the monitoring party's external tracking ID.

When the monitored subscriber is a Route Point, the *Route Point Call Added* and *ACD Call Added* event contains the “new” external tracking ID (CS1). No event is issued with the “old” external tracking ID (CS2).

When the monitored subscriber is an ACD, the *ACD Call Added* event contains the “old” external tracking ID (CS2). The *ACD Call Updated* event will always carry the updated external tracking ID for the call associated to the monitoring party (CS1). The call ID in the *ACD Call Updated* event remains the same.

When the monitored subscriber is an agent, the *Call Received* and *Call Answered* event will contain the “old” external tracking ID (CS2) but the *Call Monitored* event will always carry the updated external tracking ID for the call associated to the monitoring party (CS1). The call ID in the *Call Monitored* event remains the same.

The external tracking ID for the monitoring party’s calls does not change from the creation of the respective calls.

5.2.16.5 Request

A remote application can issue this request by sending a *MonitorNextCallRequest*. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the monitoring subscriber.
<i>address</i>	Address	Yes	The address of the monitored subscriber.
<i>location</i>	ClickToDialLocationEnum	No	When present, specifies the location of the calling user to ring for the call. When not present, locations are alerted per the calling user’s profile on Cisco BroadWorks.

Parameter	Parameter Type	Required	Description
<i>locationAddress</i>	NonEmptyToken	No	<p>Specific address of location to ring. This element should only be present if the <i>location</i> element is present and set to "BroadWorksAnywhere", "SharedCallAppearance", or "Mobility".</p> <p>When the <i>location</i> element is set to "BroadWorksAnywhere", the <i>locationAddress</i> is optional and contains a directory number. The directory number may be in E.164 format or it may just contain digits. When not present, all BroadWorks Anywhere locations are alerted. The following are valid directory numbers for the <i>locationAddress</i> element:</p> <ul style="list-style-type: none"> ▪ 9725551000 ▪ +19725551000 <p>When the <i>location</i> element is set to "SharedCallAppearance", the <i>locationAddress</i> is optional and contains a Cisco BroadWorks address of record. The address of record can be user@domain for SIP locations or user for MGCP locations. When not present, all Shared Call Appearance locations are alerted. The following are valid address of records for the <i>locationAddress</i> element:</p> <ul style="list-style-type: none"> ▪ 9725551000@broadsoft.com ▪ 9725551000 <p>When the <i>location</i> element is set to "Mobility", the <i>locationAddress</i> is optional and contains a directory number. The directory number may be in E.164 format or it may just contain digits. When not present, all BroadWorks Mobility locations are alerted. The following are valid directory numbers for the <i>locationAddress</i> element:</p> <ul style="list-style-type: none"> ▪ 9725551000 ▪ +19725551000

5.2.16.6 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>callStartInfo</i>	CallId	No	Contains the call ID and external tracking ID of the new call. It is only present when the statusCode is set to "201".

5.2.16.7 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111000	Invalid location value <location>.
400	111001	Cannot find location <locationAddress>.
400	111002	The <i>locationAddress</i> not allowed for location <location>.
400	111003	User is not assigned the <service name> service or there is no location assigned to this service.
400	111004	User does not have a primary endpoint.
400	110820	Blocked by Remote Office.
400	110552	User is not assigned the Directed Call Pickup with Barge-in service.
400	110760	User is not assigned the Call Center Monitoring service.
400	110761	Monitor next call already in progress for the target.
400	100006	Endpoint state is not valid.
400	110581	Emergency call is present.
400	110582	Blocked by prepaid service.
400	110580	User has no charge address.
400	110646	Not allowed to have more than one call outside the conference.
400	100019	Blocked by maximum simultaneous calls policy.
400	110011	Answer confirmation in progress.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Address errors, see section 5.1.3.2 Address Error .		

5.2.17 Last Number Redial

5.2.17.1 Description

This request is used to originate a call session from the specified subscriber using the last address that was dialed. The subscriber must have the Last Number Redial service assigned to use this request.

Cisco BroadWorks performs basic validation of the request and if it is accepted, it assigns a call ID and an external tracking ID to the connection and returns it in the response. It then initiates a call with the endpoint of the subscriber that initiated the dial. Cisco BroadWorks alerts the subscriber's endpoint prompting the subscriber to go off-hook. In some cases, the endpoint may support protocol extensions to immediately go off-hook without requiring a subscriber action.

Once the subscriber endpoint has gone off-hook, then the call is resumed by Cisco BroadWorks.

The same state and topology changes described for the Dial request are applicable to this request. For more information, see section [5.2.1 Dial](#).

5.2.17.2 Request

A remote application can issue this request by sending a LastNumberRedialRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the calling subscriber.
<i>location (AS only)</i>	ClickToDialLocationEnum	No	When present, specifies the location of the calling user to ring for the call. When not present, locations are alerted per the calling user's profile on Cisco BroadWorks.
<i>locationAddress (AS only)</i>	NonEmptyToken	No	<p>Specific address of location to ring. This element should only be present if the "location" element is present and set to "BroadWorksAnywhere", "SharedCallAppearance", or "Mobility".</p> <p>When the "location" element is set to "BroadWorksAnywhere", the locationAddress is optional and contains a directory number. The directory number may be in E.164 format or it may just contain digits. When not present, all Cisco BroadWorks Anywhere locations are alerted. The following are valid directory numbers for the locationAddress element:</p> <ul style="list-style-type: none"> ▪ 9725551000 ▪ +19725551000 <p>When the "location" element is set to "SharedCallAppearance", the locationAddress is optional and contains a Cisco BroadWorks address of record. The address of record can be user@domain for SIP locations or user for MGCP locations. When not present, all Shared Call Appearance locations are alerted. The following are valid address of records for the locationAddress element:</p> <ul style="list-style-type: none"> ▪ 9725551000@broadsoft.com ▪ 9725551000 <p>When the "location" element is set to "Mobility", the locationAddress is optional and contains a directory number. The directory number may be in E.164 format or it may just contain digits. When not present, all BroadWorks Mobility locations are alerted. The following are valid directory numbers for the locationAddress element:</p> <ul style="list-style-type: none"> ▪ 9725551000 ▪ +19725551000

5.2.17.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>callStartInfo</i>	CallStartInfo	No	Contains the call ID and external tracking ID of the new call. It is only present when the statusCode is set to "201".

5.2.17.4 Associated Events

The Last Number Redial produces the same event as the Dial request.

5.2.17.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111000	Invalid location value <location>.
400	111001	Cannot find location <locationAddress>.
400	111002	The <i>locationAddress</i> not allowed for location <location>.
400	111003	User is not assigned the <service name> service or there is no location assigned to this service.
400	111004	User does not have a primary endpoint.
400	110820	Blocked by Remote Office.
400	110720	User is not assigned the Last Number Redial service.
400	100006	Endpoint state is not valid.
400	110581	Emergency call is present.
400	110582	Blocked by prepaid service.
400	110580	User has no charge address
400	100019	Blocked by maximum simultaneous calls policy.
400	110646	Not allowed to have more than one call outside the conference.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.2.18 Retrieve Call List

This request is used to get a list of all the call IDs for the specified subscriber. Note that the call IDs of detached calls are not returned.

5.2.18.1 Request

A remote application can issue this request by sending a RetrieveCallInfoListRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID for which the list is requested.

5.2.18.2 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>callInfo</i>	CallInfoList	No	Information for each maintained by the subscriber. Only present when the statusCode is set to "200". For more information, see section 8 Type Definitions .

5.2.18.3 Associated Events

There are no events associated to this request.

5.2.18.4 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.2.19 Retrieve Call Details

This request is used to retrieve the details of a call. The call must be in the *Alerting*, *Active*, *Held*, or *Remote Held* state.

5.2.19.1 Request

A remote application can issue this request by sending a RetrieveCallDetailsRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID associated with the requested call.
<i>callId</i>	CallId	Yes	The call ID of the requested call.

Parameter	Parameter Type	Required	Description
<i>allowDetached</i>	EmptyContent	No	If present and true, returns the details of the call if it is already detached.

5.2.19.2 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	No	The detail of the maintained by the subscriber. Only present when the statusCode is set to "200". For more information, see section 8 Type Definitions .

5.2.19.3 Associated Events

There are no events associated to this request.

5.2.19.4 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	101002	Call <callId> not found.
400	101004	Call <callId> is in the detached state.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.2.20 Call Me Now (**AS only**)

5.2.20.1 Description

This request is used to set up a call session between an external party (possibly a non-Cisco BroadWorks subscriber), which can be reached as the specified destination, and the specified subscriber. This request is typically invoked in the context of “call-me-now” link that can be added to a subscriber’s e-mail signature. The person receiving the e-mail can click and follow the link to a web application where they can enter their own address (for example, phone number). The web application can then invoke this request by specifying the entered address (that is, the destination) and the *subscriber ID*, which would typically be encoded as part of the link.

This request is invoked anonymously on behalf of the subscriber and does not require authentication. Any credentials included with the request are ignored. However, the address parameter is screened per the subscriber’s profile for the Call Me Now service and outgoing calls.

In the context of processing the request, Cisco BroadWorks performs basic validation of the request and, if it is accepted, initiates a call to the specified destination. Once the external party answers the call, the call is redirected to the subscriber. As a service option, the external party can be prompted to enter a passcode. When this is the case, the passcode is returned in the response to the call me now request.

The following diagram provides information on the state and topology changes involved. In this example, the external party is a Cisco BroadWorks subscriber that is defined in the same group or enterprise.

Pre-conditions: Subscriber S1 and subscriber S2 are in the *Idle* state.

Request: The remote application sends a CallMeNow request for subscriber S1 with the destination phone number of S2. When BroadWorks receives the request, it first validates it and, if it is valid, creates the call C1 and Call Session S1. Subscriber S2 is alerted first, and upon answer, the Call Session S1 is redirected to Subscriber S1. A call C2 is created for S1 and the *callId* and *externalTrackingId* are included in the call events sent to the remote application for Subscriber S1.

Post-conditions: Call C1 is in the Active state, and C2 is in the Alerting state.



Figure 72 Call Me Now

5.2.20.2 Request

A remote application can issue this request by sending a CallMeNowRequest. A CallMeNowRequest contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the calling user.
<i>Address</i>	Address	Yes	Address to call.
<i>transactionId</i>	String	No	When present, the <i>transactionId</i> parameter is captured in the originating call detail record (CDR) created for the Call-Me-Now call from the target user to the external party. It can be used for various purposes such as capturing an identifier of the application or external party that is invoking the Call Me Now service. It is an optional parameter and any string value up to 128 characters long is valid.

5.2.20.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>calMeNowStartInfo</code>	CallMeNowStartInfo	No	Contains the passcode to use for the Call Me Now call. The passcode is only included when the Call Me Now service's answer confirmation option set to Passcode.

5.2.20.4 Associated Events

The events sent to the remote application for the call to Subscriber S1 are the same as the events sent when the subscriber receives a call.

In the case where the external party is a Cisco BroadWorks subscriber, then the events sent to the remote application for that subscriber are the same as the events sent when the subscriber receives a call. If the subscriber (that is, external party) is also defined in the same group or enterprise, then a *Call Transferred* event is also sent when the subscriber answers the incoming call.

5.2.20.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110970	Target user does not have the Call Me Now service or the service is disabled.
400	110971	Call Me Now calls may not be placed to an emergency or repair address.
400	110972	Call Me Now calls may not be to an address of the target user.
400	110973	Call Me Now call has been rejected by screening.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

For Address errors, see section [5.1.3.2 Address Error](#).

5.2.21 Record (AS only)

5.2.21.1 Description

This request is used by a remote application to record a call. The user must have the Call Recording service to use this command.

Receiving a success response to the command does not guarantee success of the record action requested; the success response only means that the command is being processed. However, receiving an error response is definitive; it means the command is not being processed because the validation checking detected conditions that prevent the command from being processed.

In addition, a *CallRecordingStartedEvent* is sent when the recording is started successfully.



The record command has additional capabilities if the user's call recording mode is *On Demand with User Initiated Start*. In this mode, the `calls/<callid>/record` URL allows the remote application to issue a command to the Application Server to record the call. Upon reception of this command in this mode, the Application Server starts the Call Recording service for the user's call.

There are situations in which the command is issued more than once during the life of a call. The following describes how the situation is handled in the *On Demand with User Initiated Start* mode.

- If the first record command came back with an error response, it is reasonable that the user would want to try the command again. In this situation, if the command is issued again, the Application Server reattempts to record the call. If the recording started successfully, the `CallRecordingStartedEvent` is sent. If not successful, `CallRecordingStartedEvent` is not sent.
- If the first record command resulted in a `CallRecordingStartedEvent` notification and the user issues the command a second time, the resulting response depends on whether the call is currently being recorded. If the user has stopped the recording prior to the second record command, then the success or failure of the reattempt is based on whether the `CallRecordingStartedEvent` is sent. If there is already an active recording, the Application Server returns a success response since the call is already being recorded.

If the user's call recording mode is *On Demand*, the `calls/<callid>/record` URL allows the remote application to issue a command to the Application Server to record the call. The response that is returned can be a success response or an error response.

There may be situations in which the command is issued more than once during the life of a call. The following describes how these situations are handled.

- If the first record command came back with an error response, it is reasonable that the user would want to try the command again. In this situation, if the command is issued again, the Application Server reattempts to record the call. If the recording started successfully, the `CallRecordingStartedEvent` is sent. If not successful, `CallRecordingStartedEvent` is not sent.
- If the first record command resulted in a `CallRecordingStartedEvent` and the user issues the command a second time, the Application Server returns a success response since the call is already being recorded.

If the record request is received and there are validation errors, an error response is returned; otherwise, a success response is returned.

If a subsequent record request is received while the recording is already underway or while processing an earlier record request, a success response is returned.

There is no state or topology change that occurs when the request is invoked.

5.2.21.2 Request

A remote application can issue this request by sending a RecordRequest. A RecordRequest contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID associated with the call to reconnect.
<code>callId</code>	CallId	Yes	The call ID of the call to record.



5.2.21.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.2.21.4 Associated Events

The *CallRecordingStartedEvent* is only sent when the recording is successfully started. The *recordingState* element is set to “Started” and the *allowedRecordingControls* element is updated according to the recording mode. When processing the record request, it is possible that the *CallRecordingStartingEvent* is sent prior to the *CallRecordingStartedEvent*. If the record request was not successful, the *CallRecordingStartedEvent* notification is not sent.

5.2.21.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110980	User is not assigned the Call Recording service.
400	110981	Recording is disabled for this call.
400	110982	Recording has failed for this call.
400	110985	Outstanding request in progress.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

For Call ID errors, see section [5.1.3.3 CallId Errors](#).

5.2.22 Retrieve

5.2.22.1 Description

This request is used to pull a call from one user’s location to a different user’s location. The target user must have a multiple appearance service such as BroadWorks Mobility, BroadWorks Anywhere, or Shared Call Appearance active.

If the Call Retrieve action is rejected (for example, the target user does not have a multiple appearance service active or does not have a call to retrieve), then a 400 error response is returned.

If this is not the case, a successful Hypertext Transfer Protocol (HTTP) POST request returns a *CallStartInfo* header with the URL of the call ID for the call created in the HTTP response. The HTTP response returned is a 200 OK.

A call can be retrieved when it is in the *Active*, *Held*, or *Remote Held* states.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: Call C1 can be in the *Active*, *Held*, or *Remote Hold* state.

Request: The remote application sends a Call Retrieve specifying target user S1.

Post-conditions:

- 1) Call C3 is created and alerts S1 location 2.
- 2) S1 location 2 rings and is answered.
- 3) Call C1 is connected to S1 location 2.
- 4) S1 location 1 is released.
- 5) Call C3 is released.
- 6) Call session CS1 is maintained.

Both calls C1 and C2 are in the *Active*, or *Remote Hold* state.

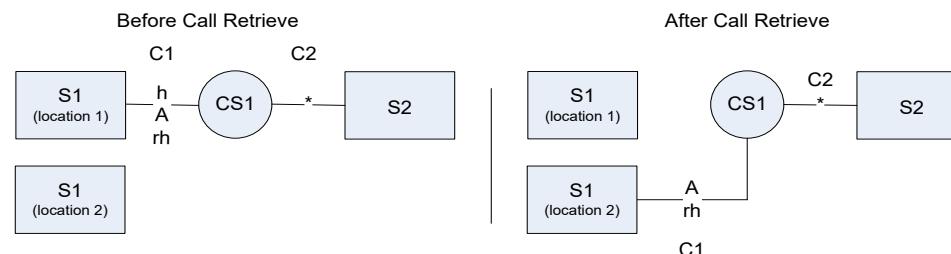


Figure 73 Retrieve

5.2.22.2 Request

A remote application can issue this request by sending a RetrieveRequest. A RetrieveRequest contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID associated with the call to reconnect.
<i>location</i>	ClickToDialLocationEnum	No	When present, specifies the location of the calling user to ring for the call. When not present, locations are alerted per the calling user's profile on Cisco BroadWorks.

Parameter	Parameter Type	Required	Description
<i>locationAddress</i>	NonEmptyToken	No	<p>Specific address of location to ring. This element should only be present if the “location” element is present and set to “BroadWorksAnywhere”, “SharedCallAppearance”, or “Mobility”.</p> <p>When the “location” element is set to “BroadWorksAnywhere”, the <i>locationAddress</i> is optional and contains a directory number. The directory number can be in E.164 format or it can just contain digits. When not present, all BroadWorks Anywhere locations are alerted. The following are valid directory numbers for the <i>locationAddress</i> element:</p> <ul style="list-style-type: none"> ▪ 9725551000 ▪ +19725551000 <p>When the “location” element is set to “SharedCallAppearance”, the <i>locationAddress</i> is optional and contains a Cisco BroadWorks address of record. The address of record can be user@domain for SIP locations or user for MGCP locations. When not present, all Shared Call Appearance locations are alerted. The following are valid address of records for the <i>locationAddress</i> element:</p> <ul style="list-style-type: none"> ▪ 9725551000@broadsoft.com ▪ 9725551000 <p>When the “location” element is set to “Mobility”, the <i>locationAddress</i> is optional and contains a directory number. The directory number may be in E.164 format or it may just contain digits. When not present, all BroadWorks Mobility locations are alerted. The following are valid directory numbers for the <i>locationAddress</i> element:</p> <ul style="list-style-type: none"> ▪ 9725551000 ▪ +19725551000

5.2.22.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>callStartInfo</i>	CallStartInfo	No	Contains the call ID and external tracking ID of the new call. It is only present when the <i>statusCode</i> is set to “201”.



5.2.22.4 Associated Events

When the Retrieve request is accepted, the following events are generated:

- Call C3 is created for the period of time where S1 location 2 is alerted. The *Call Originating*, *Call Updated* events are generated once the location is alerted. The *Call Released* event is generated once the call is answered at the specified location.
- If the Retrieve request was successful, then S1 location 2 is now linked to C1 and a *Call Updated* event is generated to reflect the change in endpoint.
- If call C1 was held, then C1 goes into *Active* and a *Call Retrieved* event is generated against subscriber S1 for call C1 indicating that the call has been put back in the *Active* state. A *Call Retrieved* event is also generated against subscriber S2 for call C2.

Note that in the case where S2 was not a Cisco BroadWorks subscriber, (that is, are Network Parties) then no event would be generated against S2 and the associated call C2.

5.2.22.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111000	Invalid location value <location>.
400	111001	Cannot find location <locationAddress>.
400	111002	The locationAddress not allowed for location <location>.
400	111003	User is not assigned the <service name> service or there is no location assigned to this service.
400	111004	User does not have a primary endpoint.
400	110820	Blocked by Remote Office.
400	111010	Target user does not have a multiple appearance service enabled.
400	111011	No call to retrieve.
400	111012	Call already assigned to location.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.2.23 Find-me/Follow-me Call Push (**AS only**)

The Find-me/Follow-me service allows a user to define a set of alerting groups. Each group contains a list of users and external devices that are simultaneously rung. The service sequentially advances through the list of alerting groups.

The Find-me/Follow-me (FMFM) Call push request allows a remote application to send a call that a user has received from a FMFM group back to the group to re-alert the members. This functionality is limited to calls that were received from the FMFM group to a Cisco BroadWorks subscriber who is part of that FMFM group.

If the request succeeds, then the call is redirected to the selected FMFM group. All the users in the alerting groups are alerted again, except for the user who pushed the call.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: Call C1 can be in the Active, Held, and Remote Held state. Subscriber S1 and S3 belong to the same alerting group.

Request: The remote application sends a Find-me/Follow-me Call Push request for S1 specifying call C1.

Post-conditions: Call C1 is released and goes into Detached state. Call session CS1 is maintained. Call C3 is created. Both C2 and C3 are in the Alerting state.

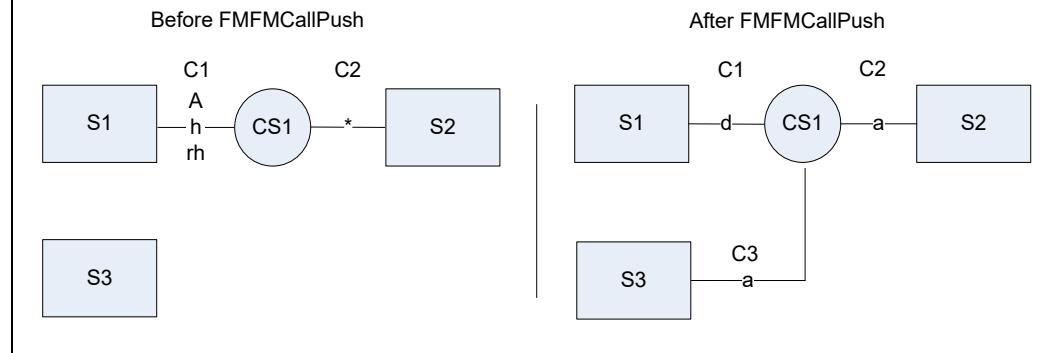


Figure 74 Find-me/Follow-me Call Push

5.2.23.1 Request

A remote application can issue this request by sending a `FindMeFollowMeCallPushRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID performing the call push.
<code>callId</code>	CallId	Yes	The callId of the call to push.

5.2.23.2 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.2.23.3 Associated Events

The following time sequence diagram illustrates the event produced when a Find-me/Follow-me Call Push request is accepted.

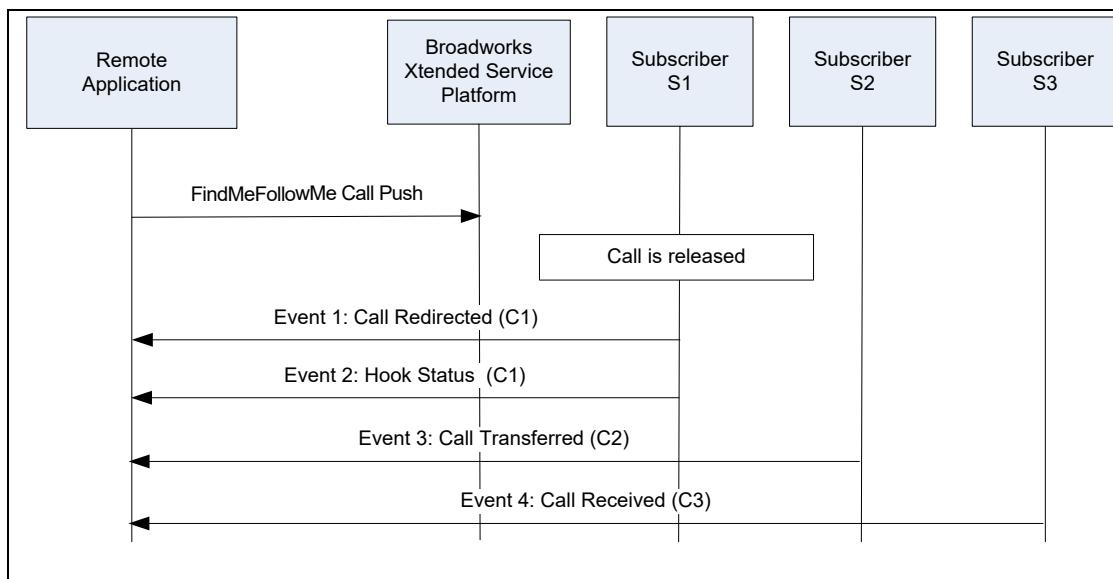


Figure 75 Find-me/Follow-me Call Push Events

Event 3 (the Call Transferred event) reports the new call state as *Alerting* with a new remote party.

5.2.23.4 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111030	User is not a member of Find-me/Follow-me group.
400	100012	User for address <address> has no primary phone number.
400	101003	Call <callId> is an Emergency call.
400	111031	Not a Find-me/Follow-me call.
400	101001	State for call <callId> is not valid.
400	101003	Call <callId> is an Emergency call.
400	101000	Call <callId> has no remote call.
400	110541	Call <callId> is an answered Push-To-Talk call.
400	100008	Remote party for call <callId> is a virtual subscriber.
400	100017	Blocked by maximum concurrent redirections policy.
401	100016	Blocked by Account/Authorization Codes service.
400	100018	Blocked by Intercept Group or Intercept User service.
400	100007	Blocked by Outgoing Calling Plan service.
400	110501	Consultation transfer in progress.



statusCode	errorCode	Summary
400	100009	Blocked by Communication Barring service.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors		

5.2.24 Allocate IMRN

5.2.24.1 Description

Mobile calls, both originating and terminating, are routed to the Voice over Internet Protocol (VoIP) domain using an IP Multimedia Routing Number (IMRN) provided by the Cisco BroadWorks Application Server. When the IMRN receives the call, the Cisco BroadWorks Telephony Application Server anchors the call to provide origination or termination services to the mobile handsets.

This request is used by a remote application to allocate an IMRN resource on Cisco BroadWorks to anchor a mobile call.

5.2.24.2 Request

A remote application can issue this request by sending an AllocateIMRNRequest. An AllocateIMRNRequest contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID associated with the IMRN request.
<i>callingPartyAddress</i>	String	Yes	<p>The address of the calling party who is originating the call.</p> <p>The following are examples of valid content for this parameter:</p> <ul style="list-style-type: none">▪ 9725551111▪ +19725551111
<i>calledPartyAddress</i>	String	Yes	<p>The dialed digits of the called party address where the call is terminated.</p> <p>The following are examples of valid content for this parameter:</p> <ul style="list-style-type: none">▪ 9725551111▪ +19725551111▪ Feature access codes▪ Service access codes such as 711, 411
<i>conferenceId</i>	String	O	A conference ID for the conference to join the Meet-Me conference.
<i>pin</i>	String	O	This is the security PIN for the Meet-Me conference to join, if the conference requires a security PIN. The length of this parameter should be a maximum of 10 digits.
<i>roomId</i>	String	O	A room ID for a collaborate room conference.

5.2.24.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
IMRNInfo	IMRInfo	No	Contains the IMRN number allocated by Cisco BroadWorks. It is only present when the statusCode is set to "201".

5.2.24.4 Associated Events

There are no events associated to this request.

5.2.24.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111020	BroadWorks Mobility service is not assigned to this user.
400	111021	BroadWorks Mobility service is not activated for this user.
400	111022	CallingPartyAddress <callingPartyAddress> not found.
400	111023	Duplicate request for same call context.
400	111024	IMRN request not allowed.
400	111025	IMRN allocation failed.
400	110040	User is not assigned the BroadWorks Anywhere service.
400	110041	BroadWorks Anywhere service is not activated for this user.
400	111250	User is not assigned the 3G/4G Continuity service.
400	111251	3G/4G Continuity service is not activated for this user.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.2.25 Executive-Assistant Initiate Call (AS only)

The Executive-Assistant Initiate Call allows the remote application to issue a command to the Application Server to make a call from the executive assistant on behalf of the executive.

The executive's address and the address to be dialed are necessary for this command.

The response that is returned can be call start information or an error response. If the Call Initiate request succeeds, a new call is placed to the address specified in the command on behalf of the executive

The following diagram provides information on the state and topology changes involved.

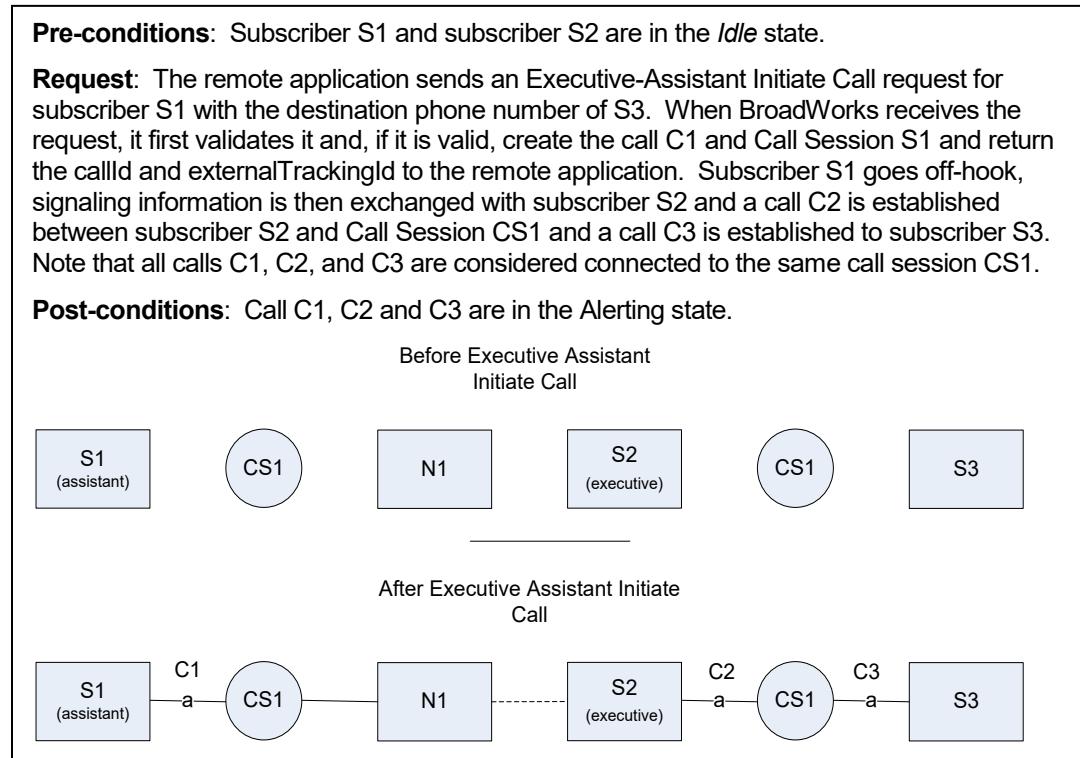


Figure 76 Executive Assistant Initiate Call

5.2.25.1 Request

A remote application can issue this request by sending an `ExecutiveAssistantInitiateCallRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID of the executive assistant.
<code>executiveAddress</code>	Address	Yes	The address of the executive
<code>address</code>	Address	Yes	Address to call.

5.2.25.2 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>callStartInfo</code>	CallStartInfo	No	Contains the call ID and external Tracking ID of the new call. It is only present when the <code>statusCode</code> is set to "201".

5.2.25.3 Associated Events

The following time sequence diagram illustrates the event produced when an Executive Assistant Initiate Call is accepted.

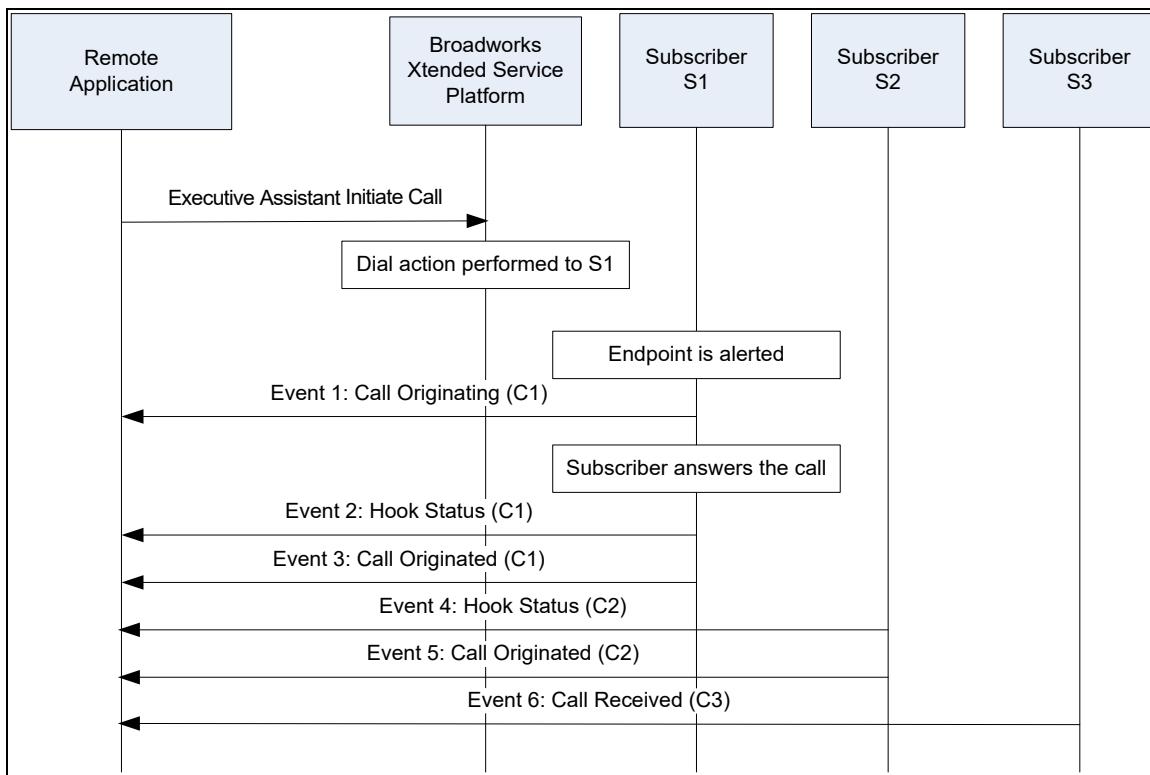


Figure 77 Events

As shown, both call C1 and C2 generate the same events.

5.2.25.4 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111110	User is not assigned the Executive-Assistant service or the service is not enabled.
400	111113	Executive user not found.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors		

5.2.26 Executive-Assistant Call Push (AS only)

The Executive-Assistant Call push allows the remote application to issue a command to the Application Server to push the call back to the executive associated with the call. The response that is returned can be a success response or an error response.

A success response is returned if the following conditions are met:

- The user must be an assistant.
- The call for the specified call ID must exist for the assistant.
- The call must be an executive call for the assistant.
- The executive call must have already been answered.

Otherwise, an error response is returned.

Note that the call can be either a filtered call answered by the assistant or a call initiated by the assistant on behalf of the executive via Executive-Assistant Call Initiation. In both scenarios, the call is referred to as an executive call for the assistant.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: Call C1 is an executive call for the assistant. All calls can be in the *Active*, *Held*, and *Remote Held* state. Subscriber S1 is the Executive-Assistant and Subscriber S2 is the Executive.

Request: The remote application sends an Executive Assistant Call Push request for S1 specifying call C1.

Post-conditions: Call C1 is released. Call session CS1 is maintained. Call C2 which previously ended at the ExecutiveAssistantEndpoint now ends at S2 AccessEndpoint. C3 is in the *Active* state if it was in the *Remote-Held* or *Active* state. C2 is in the *Active* state if it was in the *Held* or *Active* state. C2 stays in the *Remote-Held* state if it was previously in this state. C3 stays in the *Held* state if it was previously in this state.

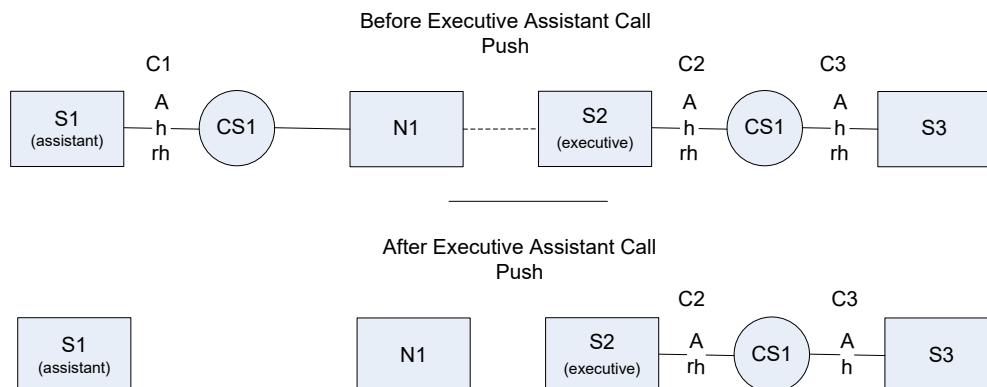


Figure 78 Executive-Assistant Call Push

5.2.26.1 Request

A remote application can issue this request by sending an ExecutiveAssistantCallPushRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID performing the call push.
<i>callId</i>	CallId	Yes	The call ID of the call to push.

5.2.26.2 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.2.26.3 Associated Events

The following time sequence diagram illustrates the event produced when an Executive-Assistant Call Push request is accepted.

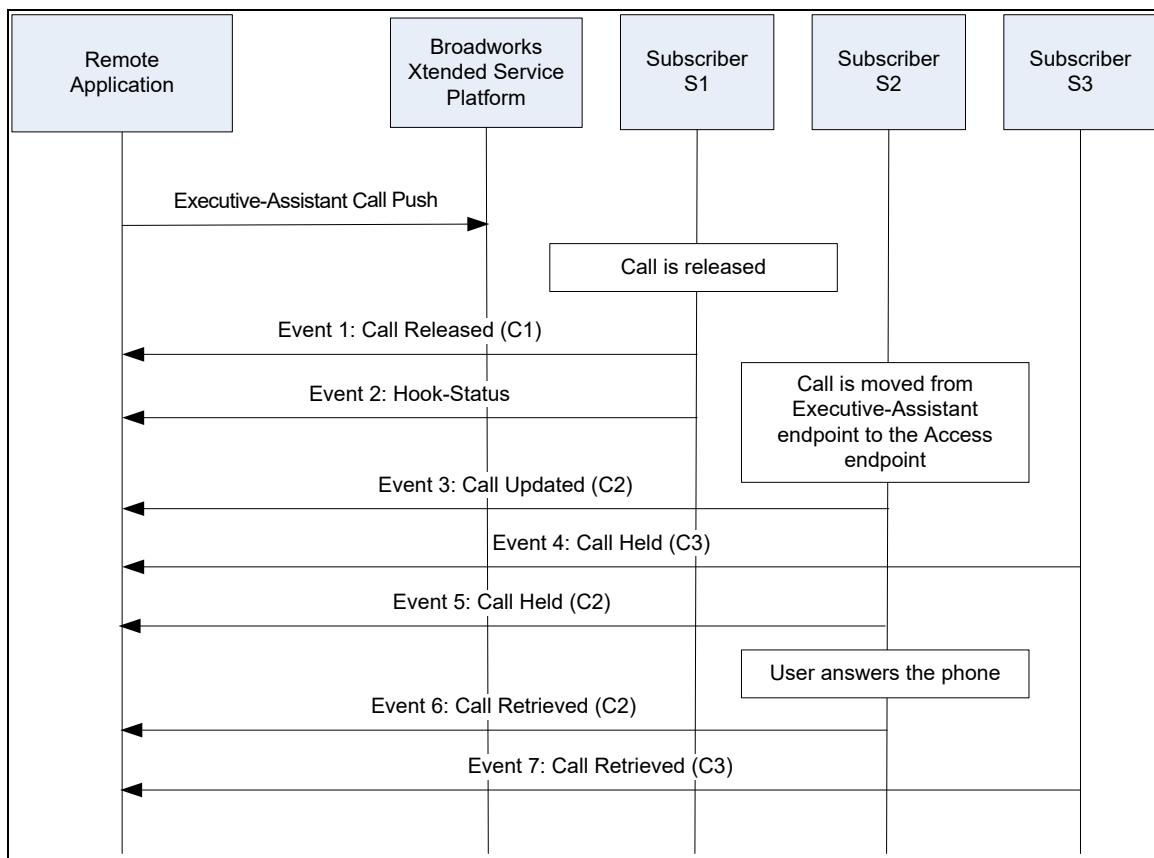


Figure 79 Executive Assistant Call Push Associated Events

5.2.26.4 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111110	User is not assigned the Executive-Assistant service or the service is not enabled.
400	101002	Call <callId> not found.
400	111111	Call <callId> is not an executive call for the assistant.
400	111112	Call <callId> has not been answered. The executive call must have been answered before initiating a call push.
For <i>System</i> errors, see section 5.1.3.5 System Errors .		
For <i>Target User</i> errors, see section 5.1.3.1 Target User Errors .		
For <i>Call ID</i> errors, see section 5.1.3.3 CallId Errors		

5.2.27 Pause Recording (AS only)

5.2.27.1 Description

This request is used by a remote application to pause the recording of a call.

The recording modes that support “pause” are: *Always with Pause/Resume*, *On Demand*, and *On Demand with User Initiated Start*. The response that is returned can be a success response or an error response.

If the pause recording request is received and there are validation errors, an error response is returned; otherwise, a success response is returned.

If a subsequent pause recording request is received while the recording is already paused or while processing an earlier pause request, a success response is returned.

If the pause was not successful, the *CallRecordingPausedEvent* notification is not sent.

There is no state or topology change that occurs when the request is invoked.

5.2.27.2 Request

A remote application can issue this request by sending a PauseRecordingRequest. A PauseRecordingRequest contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID associated with the recorded call.
<i>callId</i>	CallId	Yes	The call ID of the call to record.

5.2.27.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.2.27.4 Associated Events

Only when the recording is successfully paused, the *CallRecordingPausedEvent* is sent. The *recordingState* element is set to “Paused” and the *allowedRecordingControls* element is updated according to the recording mode.

5.2.27.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110980	User is not assigned the Call Recording service.
400	110981	Recording is disabled for this call.
400	110982	Recording has failed for this call.
400	110983	User recording mode does not support %command%.
400	110984	Recording is not active.
400	110985	Outstanding request in progress.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.2.28 Resume Recording (AS only)

5.2.28.1 Description

This request is used by a remote application to resume the recording of a call. The recording modes that support “resume” are *Always with Pause/Resume*, *On Demand*, and *On Demand with User Initiated Start*. The response that is returned can be a success response or an error response.

If the resume recording request is received and there are validation errors (see the following error codes), an error response is returned; otherwise, a success response is returned.

If a subsequent resume recording request is received while the recording is already underway or while processing an earlier resume request, a success response is returned.

There is no state or topology change that occurs when the request is invoked.

5.2.28.2 Request

A remote application can issue this request by sending a *ResumeRecordingRequest*. A *ResumeRecordingRequest* contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID associated with the call to record.
<i>callId</i>	CallId	Yes	The call ID of the call to record.

5.2.28.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.2.28.4 Associated Events

The *CallRecordingResumedEvent* is only sent when the recording is successfully resumed. The *recordingState* element is set to “Started” and the *allowedRecordingControls* element is updated according to the recording mode.

If the resume was not successful, the *CallRecordingResumedEvent* notification is not sent.

5.2.28.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110980	User is not assigned the Call Recording service.
400	110981	Recording is disabled for this call.
400	110982	Recording has failed for this call.
400	110983	User recording mode does not support %command%.
400	110984	Recording is not active.
400	110985	Outstanding request in progress.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.2.29 Stop Recording (**AS only**)

5.2.29.1 Description

This request is used by a remote application to stop the recording of a call.

The only recording mode that supports “stop” is *On-Demand with User Initiated Start*. The response that is returned can be a success response or an error response.

If the stop recording request is received and there are validation errors (see the following error codes), an error response is returned; otherwise, a success response is returned.

If a subsequent stop recording request is received while the recording is already stopped or while processing an earlier stop request, a success response is returned.

There is no state or topology change that occurs when the request is invoked.

5.2.29.2 Request

A remote application can issue this request by sending a StopRecordingRequest. A StopRecordingRequest contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID associated with the call to record.



Parameter	Parameter Type	Required	Description
<i>callId</i>	CallId	Yes	The call ID of the call to record.

5.2.29.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.2.29.4 Associated Events

The *CallRecordingStoppedEvent* is only sent when the recording is successfully stopped. The *recordingState* element is removed and the *allowedRecordingControls* element is updated according to the recording mode. In addition, a *reason* element with “User Requested” is included.

If the stop was not successful, the *CallRecordingStoppedEvent* notification is not sent.

5.2.29.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110980	User is not assigned the Call Recording service.
400	110981	Recording is disabled for this call.
400	110982	Recording has failed for this call.
400	110983	User recording mode does not support %command%.
400	110984	Recording is not active.
400	110985	Outstanding request in progress.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

For Call ID errors, see section [5.1.3.3 CallId Errors](#).

5.2.30 Set Client Session Info (AS only)

5.2.30.1 Description

This request is used by a remote application to explicitly associate the client session info with the existing call. Upon receiving this command, Cisco BroadWorks stores the client session info against the call session and triggers notifications for both ends of the call session. Notifications are sent over the SIP and/or CTI interfaces.

There is no state or topology change that occurs when the request is invoked.

5.2.30.2 Request

A remote application can issue this request by sending a SetClientSessionInfoRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID associated with the call.
<i>callId</i>	CallId	Yes	The call ID of the call for which to set the client session Info.
<i>Info</i>	String	Yes	This is an opaque information provided by the client for the Application Server to correlate this information with the call session. The Application Server stores this information against the call session and triggers notifications for both ends of the call session over the SIP or CTI interface.

5.2.30.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.2.30.4 Associated Events

The *CallClientSessionInfoUpdateEvent* is sent when the client session info is successfully set.

5.2.30.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	101001	State for call <callId> is not valid.
400	101000	Call <callId> has no remote call.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.2.31 Retrieve Hook Status (AS only)

5.2.31.1 Description

This request is used by a remote application to get the hook status for a subscriber.

There is no state or topology change that occurs when the request is invoked.

5.2.31.2 Request

A remote application can issue this request by sending a RetrieveHookStatusRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID associated with the call.

5.2.31.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>hookStatus</i>	HookStatus	No	The detail maintained by the subscriber. Only present when the statusCode is set to "200". For more information, see section 8 Type Definitions .

5.2.31.4 Associated Events

There are no events associated to this request.

5.2.31.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	100003	User not found.
400	100015	User is a virtual subscriber.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.3 Conference Requests

5.3.1 Conference Start

5.3.1.1 Description

This request is used to initiate a conference. The conference controller must have the N-Way Call or Three-Way Call service to use this request. For more information on these services, see the *Cisco BroadWorks Service Guide* [3].

A conference is established by joining two or more existing call sessions at the conference controller. A subscriber can host only one single conference. Once the conference is established, it is in the *Active* state.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: Call C1 and C3 can be in the Alerting, Active, Held or Remote Held state with personality Originator or Terminator. However, there can only be one Alerting call with Terminator personality.

Request: The remote application sends a Conference Start request for S2.

Post-conditions: Call C1 and C3 are now part of a conference and remains in the same state as they were prior to the conference except if they were in the Held or Alerting state, in which case they are automatically transited to Active state. The conference state is in the Active state.

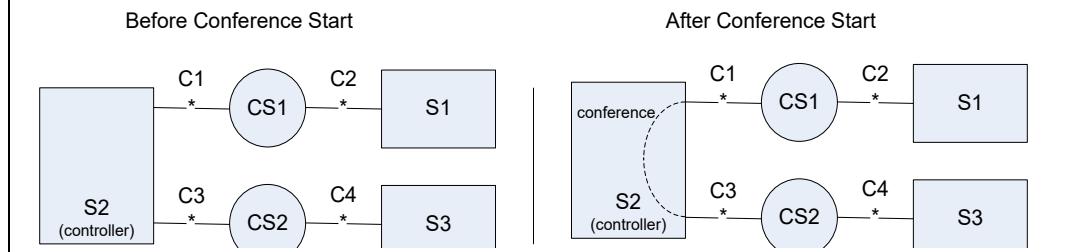


Figure 80 Conference Start

The following table summarizes how the C1 and C3 states are affected by the conference start request.

C1 or C3 State Before Conference Start	Personality	C1 or C3 State After Conference Start
Alerting	Terminator	Active
Alerting	Originator	Alerting
Held	Originator or Terminator	Active
Active	Originator or Terminator	Active
Remote Held	Originator or Terminator	Remote Held

5.3.1.2 Request

A remote application can issue this request by sending a ConferenceStartRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the conference controller.
<i>conference</i>	Conference	Yes	The list of the conference participants to include in the conference. Only the call ID parameter of the conference participants needs to be completed. The remaining data inside the conference is ignored for this request.

5.3.1.3 Response

The response contains the following parameters.

Parameter	Parameter Type	Required	Description
<code>statusCode</code>	StatusCode	Yes	The response code. Set to "200" when the request is successfully accepted, otherwise set to one of the statusCodes listed in section 5.3.1.5 Error Cases .
<code>reason</code>	Reason	Yes	The response reason. Set to "OK" when the request is successfully accepted, otherwise set to one of the reasons listed in section 5.3.1.5 Error Cases .

5.3.1.4 Associated Events

When the Conference Start request is performed, a Call Updated event is generated for each associated call followed by a ConferenceStarted event. The Call Updated event is simply used to notify that the call endpoint is no longer associated with the conference. For calls that were in the *Held* state prior to the Conference Start request, a *Call Retrieved* event is generated. For calls that were in the *Alerting* state prior to the Conference Start request, a *Call Answered* event is generated.

5.3.1.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	101007	Call <callId> does not have the expected session ID of <sessionId>.
400	101008	Invalid call IDs.
400	110010	Answer confirmation in progress for call <callId>.
400	100006	Endpoint state is not valid.
400	110110	Emergency escalation in progress.
400	110501	Consultation transfer in progress.
400	110641	Conference release in progress.
400	110642	Conference already present.
400	110581	Emergency call is present.
400	110650	User is not assigned the Three-Way Calling or N-Way Calling service.
400	110643	Number of participants <numberParticipants> exceeds maximum of <maxValue>.
400	101002	Call <callId> not found.
400	100022	Blocked by Zone Calling Restrictions.
400	100021	Endpoint state for call <callId> is not valid.
400	110644	Calls are not in proper state. Calls must be in the <i>Alerting</i> , <i>Active</i> , <i>Held</i> , or Remote Held state with a personality of Originator or Terminator. Note that there can only be one Alerting call with Terminator personality.
400	101001	State for call <callId> is not valid.
400	100026	Blocked by Session Admission Control group.

statusCode	errorCode	Summary
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.3.2 Conference Release

5.3.2.1 Description

This request is used to terminate an existing conference. All calls in the conference are released.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: Two calls joined in a conference by S2. The conference state can be Active, Held or Release. It cannot be in the Idle state.

Request: The remote sends a Conference Release request for S2.

Post-conditions: Call C1 and Call C3 are terminated with their corresponding Call Sessions CS1 and CS2.

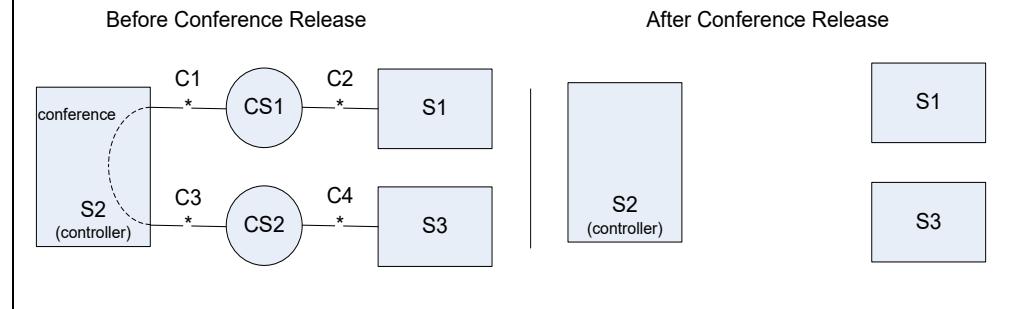


Figure 81 Conference Release State Changes

5.3.2.2 Request

A remote application can issue this request by sending a ConferenceReleaseRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
subscriberId	SubscriberId	Yes	The subscriber ID of the conference controller.

5.3.2.3 Response

The response contains the following parameters.

Parameter	Parameter Type	Required	Description
statusCode	StatusCode	Yes	The response code. Set to "200" when the request is successfully accepted, otherwise, set to one of the statusCodes listed in section 5.3.2.5 Error Cases .
reason	Reason	Yes	The response reason. Set to "OK" when the request is successfully accepted, otherwise, set to one of the reasons listed in section 5.3.2.5 Error Cases .

5.3.2.4 Associated Events

When the Conference Release request is executed, a *Conference Released* event is generated along with associated *Hook Status* events.

5.3.2.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110640	No conference present.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.3.3 Conference Hold

5.3.3.1 Description

This request is used to put a conference on hold. The subscriber who initiated this request is put on hold while other participants in the conference can still communicate.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: Two calls joined in a conference by S2. The conference state is Active.

Request: The remote application sends a Conference Hold for S2.

Post-conditions: The conference is changed to the Held state. C2 and C4 states are unchanged.

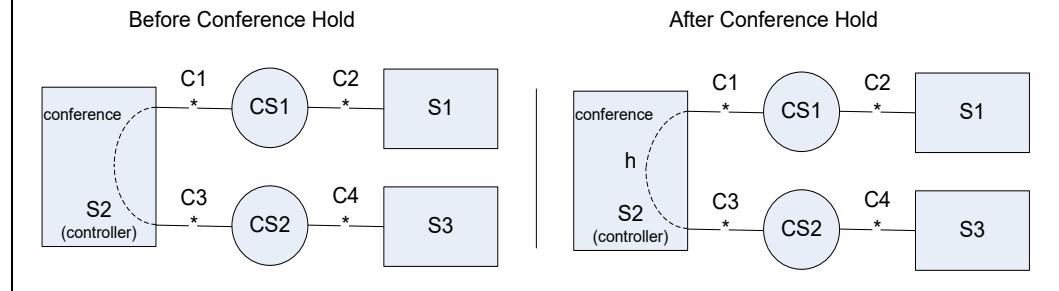


Figure 82 Conference Hold State Changes

5.3.3.2 Request

A remote application can issue this request by sending a ConferenceHoldRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the conference controller.

5.3.3.3 Response

The response contains the following parameters.

Parameter	Parameter Type	Required	Description
<i>statusCode</i>	StatusCode	Yes	The response code. Set to "200" when the request is successfully accepted. Otherwise set to one of the statusCodes listed in section 5.3.3.5 Error Cases .
<i>reason</i>	Reason	Yes	The response reason. Set to "OK" when the request is successfully accepted. Otherwise set to one of the reasons listed in section 5.3.3.5 Error Cases .

5.3.3.4 Associated Events

When the Conference Hold request is performed, a *Conference Held* is generated.

5.3.3.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110647	Conference is already held.
400	110640	No conference present.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.3.4 Conference Retrieve

5.3.4.1 Description

This request is used to resume a held conference.

The following diagram provides information on the state and topology changes involved.

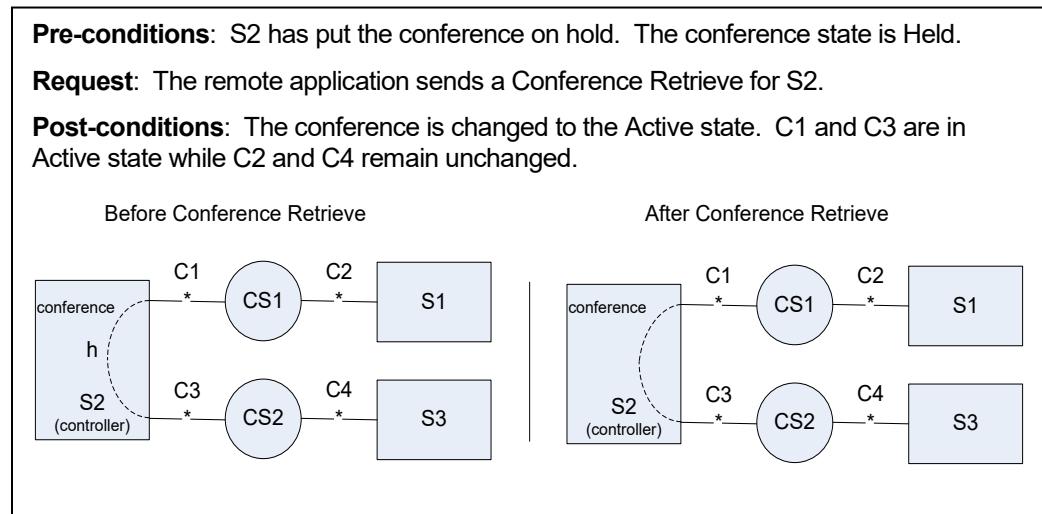


Figure 83 Conference Retrieve State Changes

5.3.4.2 Request

A remote application can issue this request by sending a ConferenceRetrieveRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the conference controller.

5.3.4.3 Response

The response contains the following parameters.

Parameter	Parameter Type	Required	Description
<i>statusCode</i>	StatusCode	Yes	The response code. Set to "200" when the request is successfully accepted. Otherwise set to one of the statusCodes listed in section 5.3.4.5 Error Cases .
<i>reason</i>	Reason	Yes	The response reason. Set to "OK" when the request is successfully accepted. Otherwise set to one of the reasons listed in section 5.3.4.5 Error Cases .

5.3.4.4 Associated Events

When the Conference Retrieve request is performed, a *Conference Retrieved* is generated.

5.3.4.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	101001	State for call <callId> is not valid.
400	100006	Endpoint state is not valid.
400	110651	Conference is already active.
400	110640	No conference present.
For <i>System</i> errors, see section 5.1.3.5 System Errors .		
For <i>Target User</i> errors, see section 5.1.3.1 Target User Errors .		

5.3.5 Conference Mute

5.3.5.1 Description

This request is used to mute a conference that is not muted by the controller.

The following diagram provides information on the state and topology changes involved.

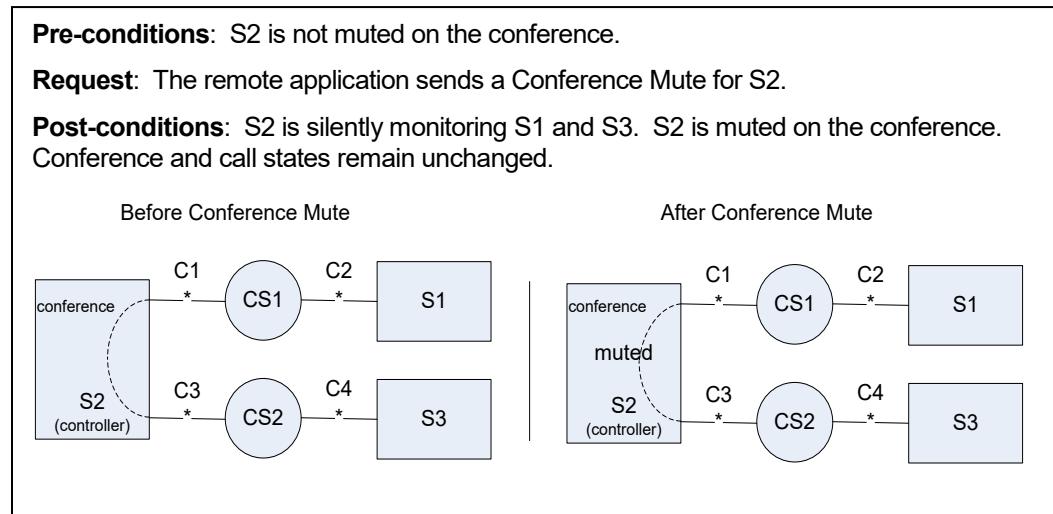


Figure 84 Conference Mute State Changes

5.3.5.2 Request

A remote application can issue this request by sending a ConferenceMuteRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the conference controller.

5.3.5.3 Response

The response contains the following parameters.

Parameter	Parameter Type	Required	Description
<i>statusCode</i>	StatusCode	Yes	The response code. Set to "200" when the request is successfully accepted. Otherwise, set to one of the statusCodes listed in section 5.3.5.5 Error Cases .
<i>reason</i>	Reason	Yes	The response reason. Set to "OK" when the request is successfully accepted. Otherwise, set to one of the reasons listed in section 5.3.5.5 Error Cases .

5.3.5.4 Associated Events

When the Conference Mute request is performed, a *Conference Muted* is generated.

5.3.5.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110640	No conference present.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.3.6 Conference Unmute

5.3.6.1 Description

This request is used to unmute a conference that is muted by the controller. Specifically, this command is used to escalate a call that is silently monitored by a supervisor to a complete barge-in.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: S2 is silently monitoring S1 and S3. S2 is muted on the conference.

Request: The remote application sends a Conference Unmute for S2.

Post-conditions: S2 is no longer muted on the conference. Conference and call states remain unchanged.

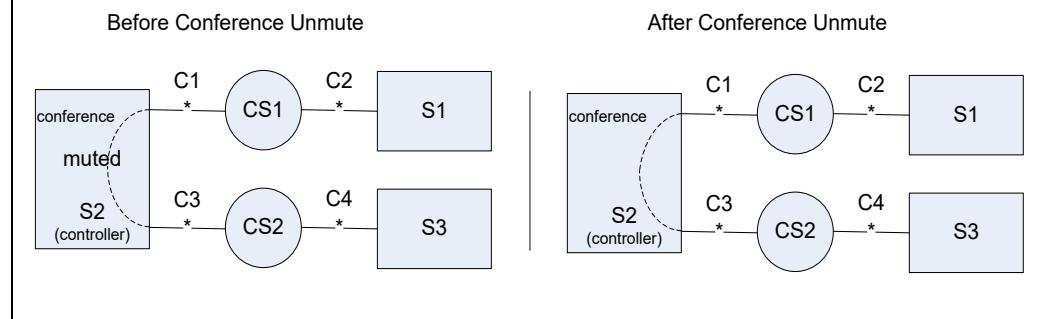


Figure 85 Conference Unmute State Changes

5.3.6.2 Request

A remote application can issue this request by sending a ConferenceUnMuteRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the conference controller.

5.3.6.3 Response

The response contains the following parameters.

Parameter	Parameter Type	Required	Description
<i>statusCode</i>	StatusCode	Yes	The response code. Set to "200" when the request is successfully accepted. Otherwise set to one of the reasons listed in section 5.3.6.5 Error Cases .
<i>reason</i>	Reason	Yes	The response reason. Set to "OK" when the request is successfully accepted, otherwise set to one of the reasons listed in section 5.3.6.5 Error Cases .

5.3.6.4 Associated Events

When the Conference Unmute request is performed, a *Conference Unmuted* is generated.

5.3.6.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110640	No conference present.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.3.7 Conference Add Participant

5.3.7.1 Description

This request is used to add a specified call as a participant to an existing conference. This specified call must belong to the subscriber who initiated this request.

The following diagram provides information on the state and topology changes involved.

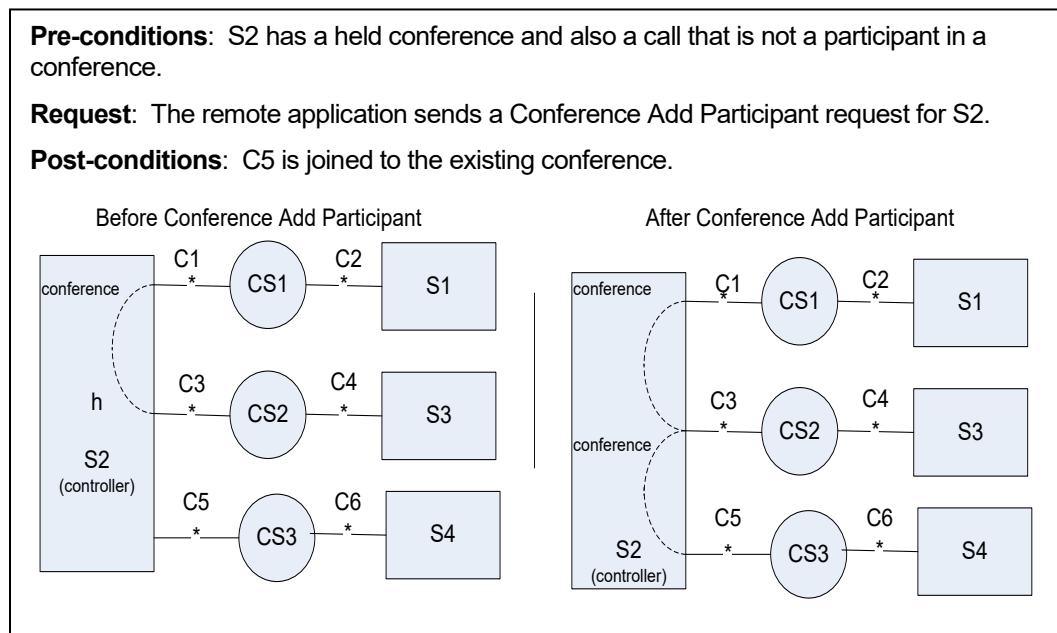


Figure 86 Conference Add Participant State Changes

5.3.7.2 Request

A remote application can issue this request by sending a ConferenceAddParticipantRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
subscriberId	SubscriberId	Yes	The subscriber ID of the conference controller.
callId	CallId	Yes	The call to be added as a participant in the subscriber's conference.

5.3.7.3 Response

The response contains the following parameters.

Parameter	Parameter Type	Required	Description
<code>statusCode</code>	StatusCode	Yes	The response code. Set to "200" when the request is successfully accepted. Otherwise, set to one of the statusCodes listed in section 5.3.7.5 Error Cases .
<code>reason</code>	Reason	Yes	The response reason. Set to "OK" when the request is successfully accepted. Otherwise, set to one of the reasons listed in section 5.3.7.5 Error Cases .

5.3.7.4 Associated Events

When the Conference Add Participant request is performed, a *Conference Updated* is generated.

For calls that were in the *Held* state prior to the request, a *Call Retrieved* event is generated.
For calls that were in the *Alerting* state prior to the request, a *Call Answered* event is generated.

5.3.7.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	101002	Call <callId> not found.
400	110648	Call <callId> is already in the conference.
400	110581	Emergency call is present.
400	100022	Blocked by Zone Calling Restrictions.
400	110640	No conference present.
400	110649	Already at maximum participants of <numberParticipants>.
400	100026	Blocked by Session Admission Control group.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Address errors, see section 5.1.3.2 Address Error .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.3.8 Mute Call

5.3.8.1 Description

This request is used to mute a call participating in a conference. When muted, no audio stream is transmitted from the muted participant associated with the call to the other participants. The video stream (if applicable) of the call is not affected.

This request does not involve any topology or call state changes.



5.3.8.2 Request

A remote application can issue this request by sending a MuteCallRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the conference controller.
<i>callId</i>	CallId	Yes	The call ID identifying the call to be muted.

5.3.8.3 Response

The response contains the following parameters.

Parameter	Parameter Type	Required	Description
<i>statusCode</i>	StatusCode	Yes	The response code. Set to "200" when the request is successfully accepted. Otherwise, set to one of the statusCodes listed in section 5.3.8.5 Error Cases .
<i>reason</i>	Reason	Yes	The response reason. Set to "OK" when the request is successfully accepted. Otherwise, set to one of the reasons listed in section 5.3.8.5 Error Cases .

5.3.8.4 Associated Events

When the Conference Mute Call request is performed, a *Conference Call Muted* is generated.

5.3.8.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	101002	Call <callId> not found.
400	110645	Call <callId> is not in the conference.
400	101001	State for call <callId> is not valid.
400	110640	No conference present.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

For Call ID errors, see section [5.1.3.3 CallId Errors](#).

5.3.9 Unmute Call

5.3.9.1 Description

This request is used to unmute a call participating in a conference. When unmuted, the audio stream can again be transmitted from the conference participant associated with the call to the other participants. In other words, this request restores the audio stream to the state it was prior to the mute request. The video stream (if applicable) is not affected.



This request does not involve any topology or call state change.

5.3.9.2 Request

A remote application can issue this request by sending an UnmuteCallRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the conference participant.
<i>callId</i>	CallId	Yes	The call ID identifying the call to be unmuted.

5.3.9.3 Response

The response contains the following parameters.

Parameter	Parameter Type	Required	Description
<i>statusCode</i>	StatusCode	Yes	The response code. Set to "200" when the request is successfully accepted. Otherwise, set to one of the statusCodes listed in section 5.3.9.5 Error Cases .
<i>reason</i>	Reason	Yes	The response reason. Set to "OK" when the request is successfully accepted. Otherwise, set to one of the reasons listed in section 5.3.9.5 Error Cases .

5.3.9.4 Associated Events

When the Conference Unmute request is performed, a *Conference Call Unmuted* is generated against the conference controller.

5.3.9.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	101002	Call <callId> not found.
400	110645	Call <callId> is not in the conference.
400	101001	State for call <callId> is not valid.
400	110640	No conference present.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

For Call ID errors, see section [5.1.3.3 CallId Errors](#).

5.3.10 Deaf Call

5.3.10.1 Description

This request is used to make deaf a call participating in a conference. When deaf, the conference audio stream from the conference controller is not transmitted to the conference participant associated with the call. However, every participant can hear the deaf party. The video stream (if applicable) to the deaf party is blacked out.

This request does not involve any topology or call state change.

5.3.10.2 Request

A remote application can issue this request by sending a DeafCallRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the conference controller.
<i>callId</i>	CallId	Yes	The call ID identifying the call to be made deaf.

5.3.10.3 Response

The response contains the following parameters.

Parameter	Parameter Type	Required	Description
<i>statusCode</i>	StatusCode	Yes	The response code. Set to “200” when the request is successfully accepted. Otherwise, set to one of the statusCodes listed in section 5.3.10.5 Error Cases .
<i>reason</i>	Reason	Yes	The response reason. Set to “OK” when the request is successfully accepted. Otherwise, set to one of the reasons listed in section 5.3.10.5 Error Cases .

5.3.10.4 Associated Events

When the Conference Deaf Participant request is performed, a *Conference Call Made Deaf* is generated against the conference controller.

5.3.10.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	101002	Call <callId> not found.
400	110645	Call <callId> is not in the conference.
400	101001	State for call <callId> is not valid.
400	110640	No conference present.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

For Call ID errors, see section [5.1.3.3 CallId Errors](#).

5.3.11 Undeaf Call

5.3.11.1 Description

This request is used to undeaf a call participating in a conference. When this request is completed, the audio stream is transmitted again to the conference participant associated with the call. The video stream (if applicable) is restored.

This request does not involve any topology or call state change.

5.3.11.2 Request

A remote application can issue this request by sending an UndeafCallRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the conference controller.
<i>callId</i>	CallId	Yes	The call ID identifying the call to be rest deaf.

5.3.11.3 Response

The response contains the following parameters.

Parameter	Parameter Type	Required	Description
<i>statusCode</i>	StatusCode	Yes	The response code. Set to "200" when the request is successfully accepted. Otherwise, set to one of the statusCodes listed in section 5.3.11.5 Error Cases .
<i>reason</i>	Reason	Yes	The response reason. Set to "OK" when the request is successfully accepted. Otherwise, set to one of the reasons listed in section 5.3.11.5 Error Cases .

5.3.11.4 Associated Events

When the Conference Undeaf Participant request is performed, a *Conference Call Made Undeaf* is generated against the conference controller.

5.3.11.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	101002	Call <callId> not found.
400	110645	Call <callId> is not in the conference.
400	101001	State for call <callId> is not valid.
400	110640	No conference present.

For System errors, see section [5.1.3.5 System Errors](#).

statusCode	errorCode	Summary
For <i>Target User</i> errors, see section 5.1.3.1 Target User Errors .		
For <i>Call ID</i> errors, see section 5.1.3.3 CallId Errors .		

5.4 Route Point Requests (**AS only**)

5.4.1 Get Route Point

5.4.1.1 Description

This request is used to get the list of calls maintained in a Route Point queue.

5.4.1.2 Request

A remote application can issue this request by sending a GetRoutePointRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>routepointId</i>	SubscriberId	Yes	The subscriber ID of the route point.

5.4.1.3 Response

The response contains the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>RoutePointQueue</i>	RoutePoint Queue	No	The detail of the Route Point queue. Only present if the statusCode was set to "200".

5.4.1.4 Associated Events

No events are associated with this request.

5.4.1.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110407	User is not a route point.
For <i>System</i> errors, see section 5.1.3.5 System Errors .		
For <i>Target User</i> errors, see section 5.1.3.1 Target User Errors .		

5.4.2 Distribute Call

5.4.2.1 Description

This request is used to offer a call queued at a Route Point to an agent. The following diagram provides information on the state and topology changes involved.

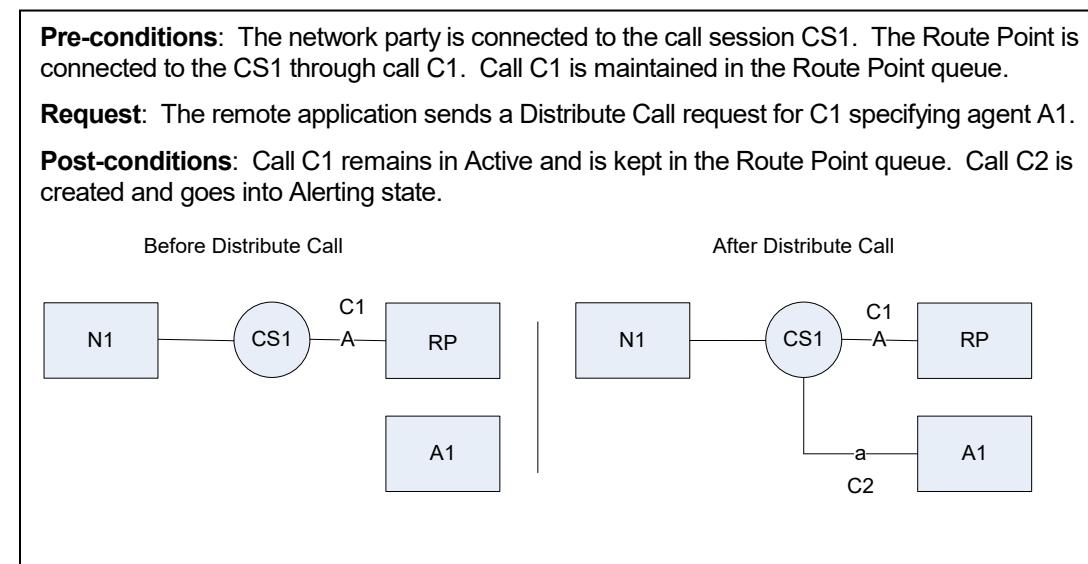


Figure 87 Distribute Call State Changes

The Route Point Distribute Call request must be used instead of the Route Point Blind Transfer request if the policies offered by the route point such as bounced call processing are to be performed. Contrary to the Route Point Blind Transfer, call C1 is maintained in the Route Point queue until the call is answered by the Agent.

A remote application can optionally add some parameters to the Distribute Call request, which are to be displayed on the agent's customer premises equipment (CPE): the Calling Line ID number and phone, the time spent in queue by this call, and the queue longest waiting time. A whisper message can also be specified. The whisper message is a call treatment that consists of the announcements played to the agent when the call is connected. The caller receives a ringing while the agent receives the whisper message based on the configuration.

The Route Point can be configured such that a ringback is played to the queued party (N1 in the diagram). However, in this case, any request to explicitly play some media (for example, Route Point Play Ringback) is rejected by Cisco BroadWorks.

For outgoing calls, this request can be performed only on answered calls.

5.4.2.2 Request

A remote application can issue this request by sending a `RoutePointDistributeCallRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>routepointId</code>	SubscriberId	Yes	The subscriber ID of the route point.
<code>callId</code>	CallId	Yes	The call ID of the call to distribute.
<code>routePointDistribute</code>	RoutePointDistribute	Yes	The detail of the distribution request.

5.4.2.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.4.2.4 Associated Events

The following time sequence diagram illustrates the event produced when a Route Point Distribute Call request is issued with no whisper message specified.

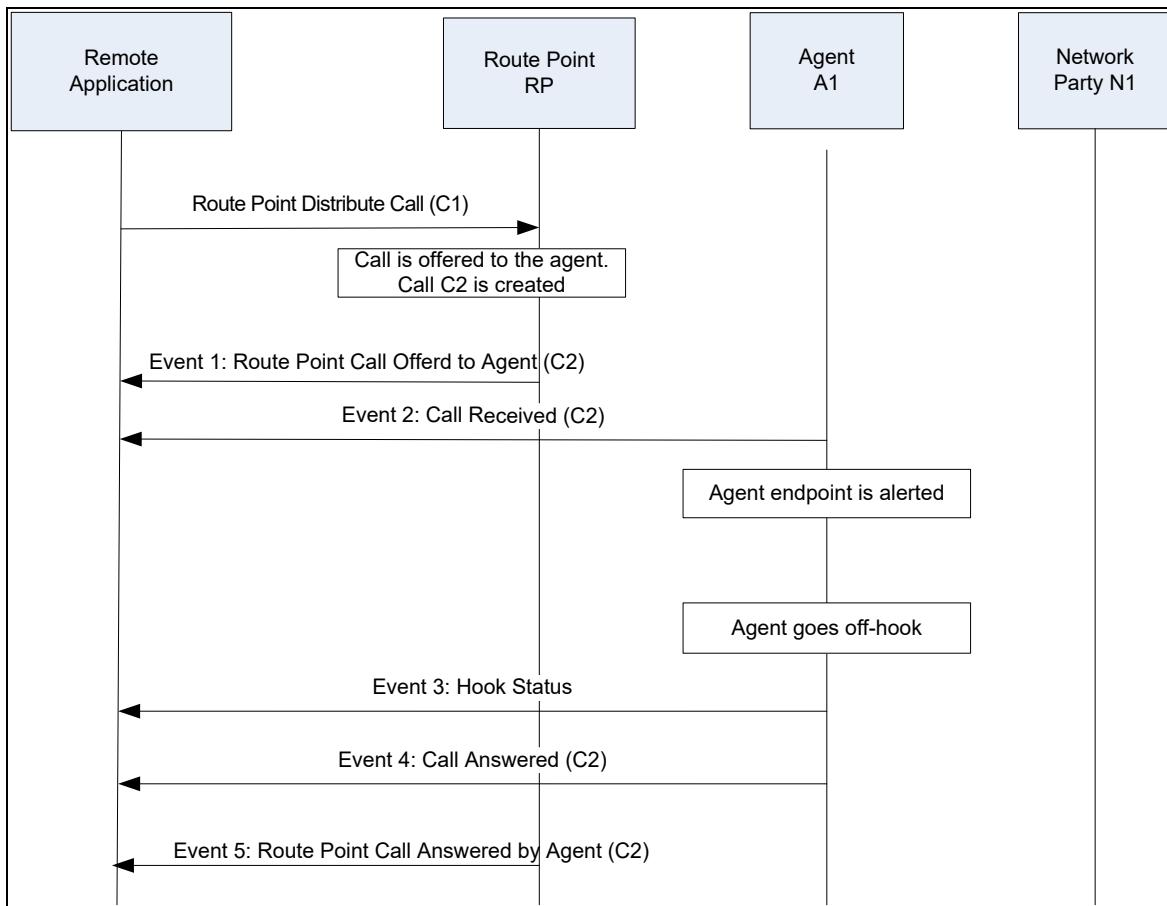


Figure 88 Route Point Distribute Call Events

If the call is not answered by the agent, then the event 3 and 4 in the diagram are replaced by the following events:

- A *Route Point Call Bounced* event against the Route Point's call (C1 in the diagram).
- A *Call Released* event against the agent's call (C2 in the diagram).

The following time sequence diagram illustrates the event produced when a Route Point Distribute Call request is issued when a whisper message specified in the request.

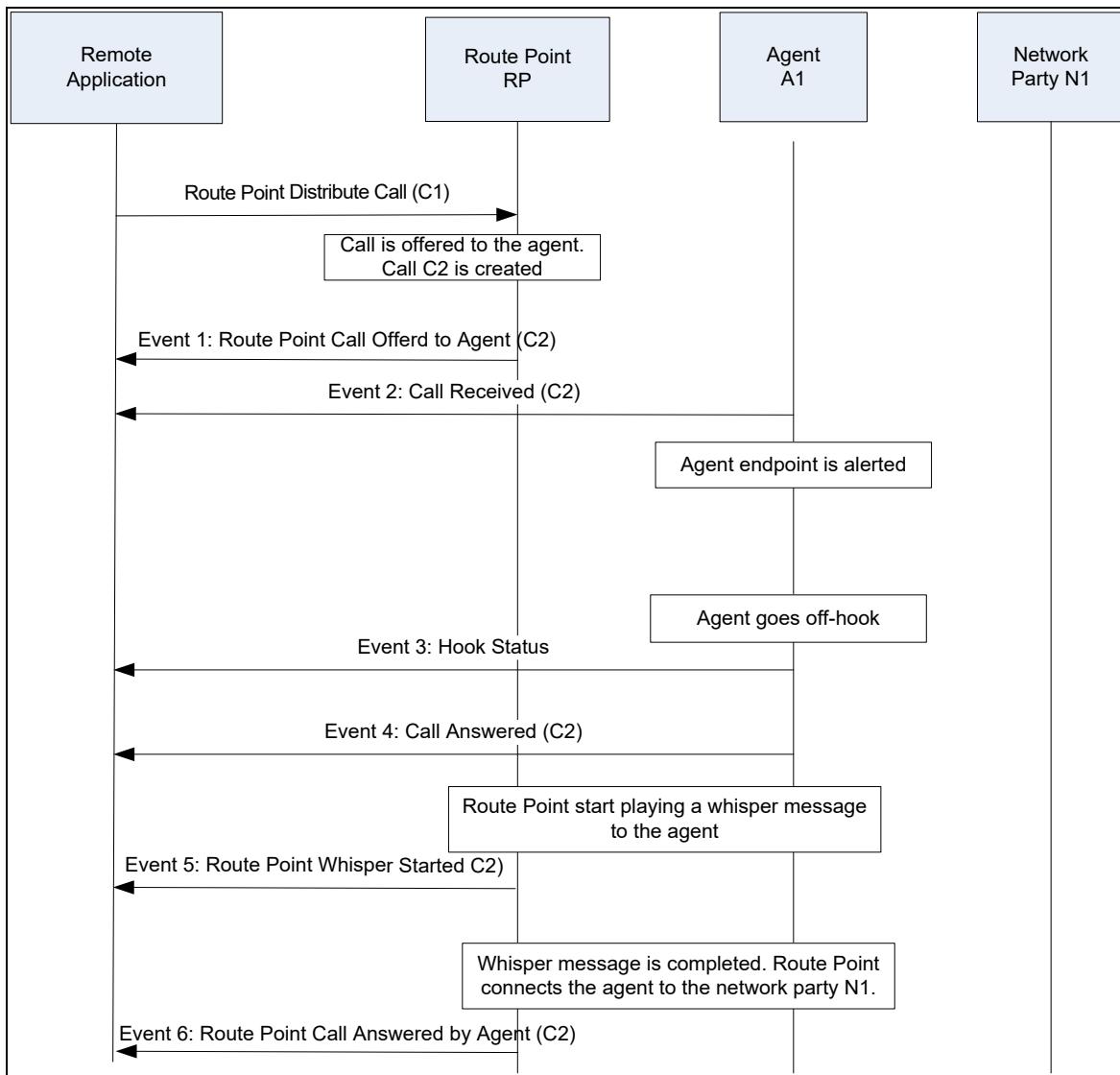


Figure 89 Route Point Distribute Call

As shown in the diagram, a Route Point Whisper Started event (Event 4) is issued when the whisper message is played to the agent. The Route Point Call Answered event (Event 5) is only sent when the whisper message is completed and the agent is connected to network party N1.

5.4.2.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110407	User is not a route point.
400	110400	Agent <userId> not found.

statusCode	errorCode	Summary
400	110108	Agent is not assigned the Call Center-Premium service.
400	101001	State for call <callId> is not valid.
400	110402	Route Point failed.

5.4.3 Play Treatment

5.4.3.1 Description

This request is used to play a treatment to a selected call in the Route Point queue.

The treatment is defined by a series of up to four audio or video URLs to be played sequentially. This sequence can be repeated up to five times. Therefore, each URL can be played five times resulting in a maximum of 20 treatments per Play Treatment request. If an MGCP digit map is specified in the request, digits can be collected while the treatment is being played.

The treatment playback can be stopped explicitly or implicitly as explained in section [4.3.6 Media Playback and Digits Collection](#). Once a treatment is stopped, no media (audio or video) is sent for the call.

This request does not involve any topology or call state changes.

5.4.3.2 Request

A remote application can issue this request by sending a RoutePointPlayTreatmentRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>routepointId</i>	SubscriberId	Yes	The subscriberId of the route point.
<i>callId</i>	CallId	Yes	The callId of the call to receive treatment.
<i>RoutePointPlayTreatment</i>	RoutePointPlayTreatment	Yes	The detail of the play request.

5.4.3.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>RoutePointPlayInfo</i>	RoutePointPlayInfo	Not applicable	Contains the playCollectId used to correlate the request with its associated event. Present only when the statusCodes is set to "200".

5.4.3.4 Associated Events

When a Route Point Play Treatment request is successfully accepted, a *Route Point Treatment Started* event is issued. When the request is completed, a *Route Point Treatment Completed* event is generated with the reason set appropriately.

To correlate these events with the request, the remote application can use the *playCollectId* returned in the response. This field is used by Cisco BroadWorks to fill the *playCollectId* field of the event associated to this Route Point Play Treatment request. This way, the remote application can easily associate the event with the request.

5.4.3.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110407	User is not a route point.
400	110402	Route Point is failed.
400	101001	State for call <callId> is not valid.
400	110411	Offered call is configured to play ringback and cannot have a treatment applied.
400	110409	Play collect ID <playCollectId> not found.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.4.4 Play Music On Hold

5.4.4.1 Description

This request is used to play Music On Hold (MOH) to a selected call in the Route Point queue.

The MOH to be played is defined through Route Point configuration. Digits can be collected while the MOH is being played if an MGCP digit map is specified in the request. However, if the MOH is played from an external source to Cisco BroadWorks, no digit collection can be performed.

The duration of the MOH to play can be optionally specified. If the duration is provided, then the MOH is discontinued when the time is reached. If the duration is not provided, the MOH is played without interruption.

When the Play Music On Hold request is successfully accepted by Cisco BroadWorks, a play-collect operation is started. A playCollectId is assigned to the operation and is returned in the response.

A remote application can update the MOH duration by sending another request for the same call. The MOH duration will be replaced by the new value and the playback timer will be reset to zero. When extending the MOH duration no new URL or MGCP digit map can be specified. The original playCollectId must be specified in the request. If the request is successfully accepted by Cisco BroadWorks, the same playCollectId will be returned in the response.

The MOH can be stopped explicitly or implicitly as explained in [4.3.6 Media Playback and Digits Collection](#).

This request does not involve any topology or call state change.

5.4.4.2 Request

A remote application can issue this request by sending a RoutePointPlayMOHRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>routepointId</i>	SubscriberId	Yes	The subscriber ID of the route point.
<i>callId</i>	CallId	Yes	The call ID of the call to receive the Music On Hold.
<i>RoutePointPlayMusicOnHold</i>	RoutePointPlayMusicOnHold	Yes	The detail of the play request.

5.4.4.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>RoutePointPlayInfo</i>	RoutePointPlayInfo	No	The RoutePointPlayInfo contains the playCollectId used to correlate the request with its associated event. Only present when the statusCode is set to "200".

5.4.4.4 Associated Events

When a Route Point Play MOH request is received, a *Route Point MOH Started* event is issued. When the request is completed, a *Route Point MOH Completed* event is generated with the reason value set appropriately.

To correlate these events with the request, the remote application can use the playCollectId returned in the response. This field is used by Cisco BroadWorks to fill the playCollectId field of the event associated to this Route Point Play MOH request. This way, the remote application can easily associate the event with the request.

5.4.4.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110407	User is not a route point.
400	110402	Route Point is failed.
400	101001	State for call <callId> is not valid.
400	110411	Offered call is configured to play ringback and cannot have a treatment applied.
400	110409	Play collect ID <playCollectId> not found.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.4.5 Play Ringback

5.4.5.1 Description

This request is used to play localized ringback to a selected call in the Route Point queue.

If an MGCP digit map is specified in the request, digits will be collected while the ringback is being played.

The duration of the ringback to play can be optionally specified. If the ringback duration is provided, then the ringback is discontinued when the time is reached. If the duration is not provided, the ringback is played without interruption.

When the Play Ringback request is successfully accepted by Cisco BroadWorks, a play-collect operation is started. A playCollectId is assigned to the operation and is returned in the response.

A remote application can update the ringback duration by sending another request for the same call. The ringback duration will be replaced by the new specified value and the playback timer will be reset to zero. When extending the ringback duration no new MGCP digit map can be specified. The original playCollectId must be specified in the request with the new duration. If the request is successfully accepted by Cisco BroadWorks, the same playCollectId will be returned in the response.

The ringback can be stopped explicitly or implicitly as explained in [4.3.6 Media Playback and Digits Collection](#).

This request does not involve any topology or call state change.

5.4.5.2 Request

A remote application can issue this request by sending a RoutePointPlayRingbackRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>routepointId</i>	SubscriberId	Yes	The subscriber ID of the route point.
<i>callId</i>	CallId	Yes	The call ID of the call to receive the ringback.
<i>RoutePointPlayRingBack</i>	RoutePointPlayRingBack	Yes	The details of the play request.

5.4.5.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>RoutePointPlayInfo</i>	RoutePointPlayInfo	No	The RoutePointPlayInfo contains the playCollectId used to correlate the request with its associated event. Only present when the statusCode is set to "200".

5.4.5.4 Associated Events

When a Route Point Play Ringback request is successfully accepted, a *Route Point Ringback Started* event is issued. When the request is completed, a *Route Point Ringback Completed* event is generated with the reason set appropriately. For more information, see section [8.8.12.2 PlayCollectResult](#).

To correlate these events with the request, the remote application can use the playCollectId returned in the response. This field is used by Cisco BroadWorks to fill the playCollectId field of the event associated to this Route Point Play Treatment request. This way, the remote application can easily associate the event with the request.

5.4.5.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110407	User is not a route point.
400	110402	Route Point is failed.
400	101001	State for call <callId> is not valid.
400	110411	Offered call is configured to play ringback and cannot have a treatment applied.
400	110409	Play collect ID <playCollectId> not found.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.4.6 Play Silence

5.4.6.1 Description

This request is used to play localized silence to a selected call in the Route Point queue.

If an MGCP digit map is specified in the request, digits will be collected while the silence is being played.

The duration of the silence to play can be optionally specified. If the silence duration is provided, then the silence is discontinued when the time is reached. If the duration is not provided, the silence is played without interruption.

When the Play Silence request is successfully accepted by Cisco BroadWorks, a play-collect operation is started. A playCollectId is assigned to the operation and is returned in the response.

A remote application can update the silence duration by sending another request for the same call. The silence duration will be replaced by the new specified value and the playback timer will be reset to zero. When extending the silence duration no new MGCP digit map can be specified. The original playCollectId must be specified in the request with the new duration. If the request is successfully accepted by Cisco BroadWorks, the same playCollectId will be returned in the response.

The silence playback can be stopped explicitly or implicitly as explained in [4.3.6 Media Playback and Digits Collection](#).

This request does not involve any topology or call state change.

5.4.6.2 Request

A remote application can issue this request by sending a RoutePointPlaySilenceRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>routepointId</i>	SubscriberId	Yes	The subscriber ID of the route point.
<i>callId</i>	CallId	Yes	The call ID of the call to receive the silence.
<i>RoutePointPlaySilence</i>	RoutePointPlaySilence	Yes	The detail of the play request.

5.4.6.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>RoutePointPlayInfo</i>	RoutePointPlayInfo	No	The RoutePointPlayInfo contains the playCollectId used to correlate the request with its associated event. Only present when the statusCode is set to "200".

5.4.6.4 Associated Events

When a Route Point Play Silence request is successfully accepted, a *Route Point Silence Started* event is issued. When the request is completed, a *Route Point Silence Completed* event is generated with the reason set appropriately. For more information, see section [8.8.12.2 PlayCollectResult](#).

To correlate these events with the request, the remote application can use the playCollectId returned in the response. This field is used by Cisco BroadWorks to fill the playCollectId field of the event associated to this Route Point Play Treatment request. This way, the remote application can easily associate the event with the request.

5.4.6.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110407	User is not a route point.
400	110402	Route Point is failed.
400	101001	State for call <callId> is not valid.
400	110411	Offered call is configured to play ringback and cannot have a treatment applied.
400	110409	Play collect ID <playCollectId> not found.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.4.7 Play Busy

5.4.7.1 Description

This request is used to play localized busy to a selected call in the Route Point queue.

The duration of the busy to play can be optionally specified. If the busy duration is provided, then the busy is discontinued when the time is reached. If the duration is not provided, the busy is played without interruption.

When the Play Busy request is successfully accepted by Cisco BroadWorks, a play-collect operation is started. A playCollectId is assigned to the operation and is returned in the response.

A remote application can update the busy duration by sending another request for the same call. The busy duration will be replaced by the new specified value and the playback timer will be reset to zero. The original playCollectId must be specified in the request with the new duration. If the request is successfully accepted by Cisco BroadWorks, the same playCollectId will be returned in the response.

The busy can be stopped explicitly or implicitly as explained in [4.3.6 Media Playback and Digits Collection](#).

This request does not involve any topology or call state change.

5.4.7.2 Request

A remote application can issue this request by sending a RoutePointPlayBusyRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>routepointId</i>	SubscriberId	Yes	The subscriber ID of the route point.
<i>callId</i>	CallId	Yes	The call ID of the call to receive the busy.
<i>RoutePointPlayBusy</i>	RoutePointPlayBusy	Yes	The detail of the play request.

5.4.7.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>RoutePointPlayInfo</i>	RoutePointPlayInfo	No	The RoutePointPlayInfo contains the playCollectId used to correlate the request with its associated event. Only present when the statusCode is set to "200".

5.4.7.4 Associated Events

When a Route Point Play Busy request is successfully accepted, a *Route Point Busy Started* event is issued. When the request is completed, a *Route Point Busy Completed* event is generated with the reason set appropriately. For more information, see section [8.8.12.2 PlayCollectResult](#).



To correlate these events with the request, the remote application can use the playCollectId returned in the response. This field is used by Cisco BroadWorks to fill the playCollectId field of the event associated to this Route Point Play Treatment request. This way, the remote application can easily associate the event with the request.

5.4.7.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110407	User is not a route point.
400	110402	Route Point is failed.
400	101001	State for call <callId> is not valid.
400	110411	Offered call is configured to play ringback and cannot have a treatment applied.
400	110409	Play collect ID <playCollectId> not found.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.4.8 Outgoing Dial

5.4.8.1 Description

This request is used to originate a call session from the Route Point to the specified destination. If the request is accepted, then Cisco BroadWorks performs the following actions:

- Assigns a call ID and an external tracking ID to the connection and returns it in the response.
- Creates the call and adds it to the Route Point queue.
- Starts a no answer timer.
- Issues a *Route Point Outgoing Call Originated* event.

The call is removed from queue if:

- The no answer timer triggers, which means that the call was not answered by the remote destination within the configured timeout value. The timer was started when the request was received.
- The call attempt failed (for example, because the remote destination is busy).
- The remote application explicitly distributes the call to an agent who answers the call.
- The callee hangs up after answering.
- The remote application performs an explicitly release of the call.

Note that if the call is answered by the remote destination while being queued, then silence is automatically played to the callee until the call is distributed and answered by an agent.

The following diagram provides information on the state and topology changes involved.

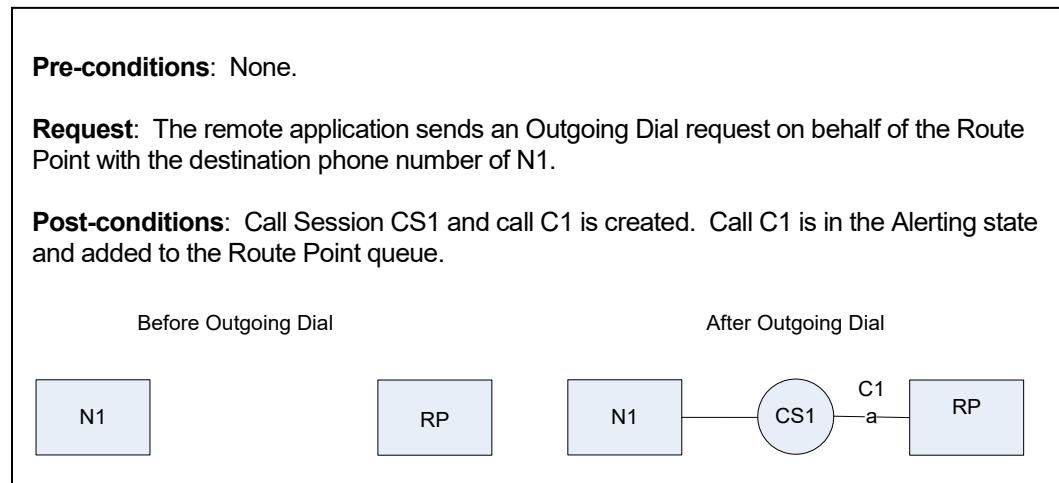


Figure 90 Outgoing Dial State Changes

5.4.8.2 Request

A remote application can issue this request by sending a `RoutePointOutgoingDialRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>routePointId</code>	SubscriberId	Yes	The subscriber ID of the route point.
<code>RoutePointOutgoingDial</code>	RoutePointOutgoingDial	Yes	The detail of the dial request.

5.4.8.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>RoutePointcallStartInfo</code>	<code>RoutePointCallStartInfo</code>	No	Contains the call ID and external tracking ID of the new call. It is only present when the <code>statusCode</code> is set to "201".

5.4.8.4 Associated Events

Many Route Point events are generated as an outgoing call progresses from being initiated to being answered. The following table lists the events generated for various cases. Each row represents a specific case and each column represents the events that are generated for the specific case. The order the events are generated is represented from left to right. For instance, in the first row, when an invalid number is dialed, then a *Route Point Call Added* event is first sent followed by a *Route Point Outgoing Call Originated* event and finally followed by a *Route Point Call Released* event (as shown in the third column) with the outgoingCallReleaseCause set. The second row describes the case where the call attempt failed (because the destination is busy for instance). In this case, a *Route Point Call Originated* event is first produced followed by a *Route Point Call Originated* event and finally followed by a *Route Point Call Released*.

Cases				Events			
	Route Point Call Added	Route Point Call Updated	Route Point Outgoing Call Originated	Route Point Call Released	Route Point Outgoing Call Answered	Route Point Call Updated	Route Point Call Abandoned
Invalid number	Generated with callId, external TrackingId and addTime set	—	Generated	Generated with outgoing Call Release Cause and remove Timeset The release Reason is set to Released WithCause	—	—	—
Failed attempts (for example, busy, invalid destination)	Generated with callId, external TrackingId and addTime set	—	Generated	Generated with outgoing Call Release Cause and remove Timeset release Reason is set to Released WithCause	—	—	—
Answer timeout	Generated with callId, external TrackingId and addTime set	Generated with remote party information	Generated	Generated with remove Timeset release Reason is set to NoAnswerTimeout	—	—	—

Cases				Events			
	Route Point Call Added	Route Point Call Updated	Route Point Outgoing Call Originated	Route Point Call Released	Route Point Outgoing Call Answered	Route Point Call Updated	Route Point Call Abandoned
Callee answered	Generated with callId, external TrackingId and addTime set	Generated with remote party information	Generated	—	Generated with Outgoing CallAnswer Timeset	Generated when CPD result is available	—
Callee hangs up before call is answered by agent	Generated with callId, external TrackingId and addTime set	Generated with remote party information	Generated	—	Generated with Outgoing CallAnswer Timeset	Generated when CPD result is available	Generated with removeTime

5.4.8.4.1 Call Progress Detection Results

The Call Progress Detection (CPD) result is used to provide information on the nature of the callee or the outcome of the call setup. Specifically, when a call session is being established, the CPD result specifies if the call was answered by a voice messaging system, a fax machine, or treated with a Special Information Tone (SIT).

The CPD result is provided in an event issued by Cisco BroadWorks. The *Route Point Call Released* event provides information for SIT. The *Route Point Call Updated* event specifies if the call was answered by a voice messaging system or a fax machine.

NOTE: Cisco BroadWorks is only able to provide an accurate CPD result based on the information received in the SIP signaling.

5.4.8.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110407	User is not a route point.
400	110404	Route point is inactive.
400	110402	Route Point is failed.
400	110405	Route point name <RPName> is not valid.
400	110406	Route point queue is full.

5.4.9 Bounce Call

5.4.9.1 Description

This request is used to bounce a call queued at a Route Point to an agent. It is only possible to bounce a call that is offered to an agent. In other words only calls that have produced a *Route Point Call Offered to Agent* event but not a *Route Point Call Answered by Agent* event yet can be bounced.

When a call is bounced, the bounced call policies are applied as explained in [4.3.1.1 Bounced Call Policy](#). Hence, two cases can be identified.

5.4.9.1.1 Case 1: Bounced Call is Put Back in Route Point Queue

In this case, Cisco BroadWorks stops alerting the agent and remains in the queue as specified by the Bounced Call Policies.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: The network party is connected to the call session CS1. The Route Point is connected to the CS1 through call C1. Call C1 is maintained in the Route Point queue. The call was offered to agent A1. Agent A1 has not answered the call C1.

Request: The remote application sends a Route Point Bounce Call request for RP specifying call C1.

Post-conditions: Call C1 remains in Active and is kept in the Route Point queue. Call C2 is released.

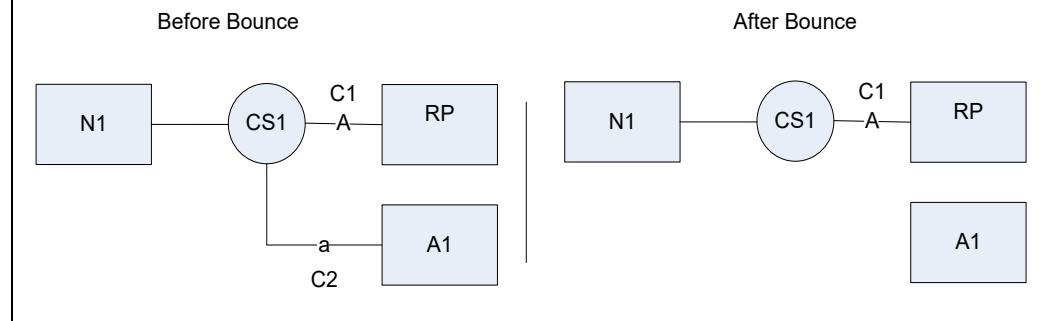


Figure 91 Bounced Call is Put Back in Route Point Queue State Changes

5.4.9.1.2 Case 2: Bounced Call is Transferred to New Destination

In this case, Cisco BroadWorks stops alerting the agent and transfer the call to a new destination as specified by the Bounced Call Policies.

The following diagram provides information on the state and topology changes involved.

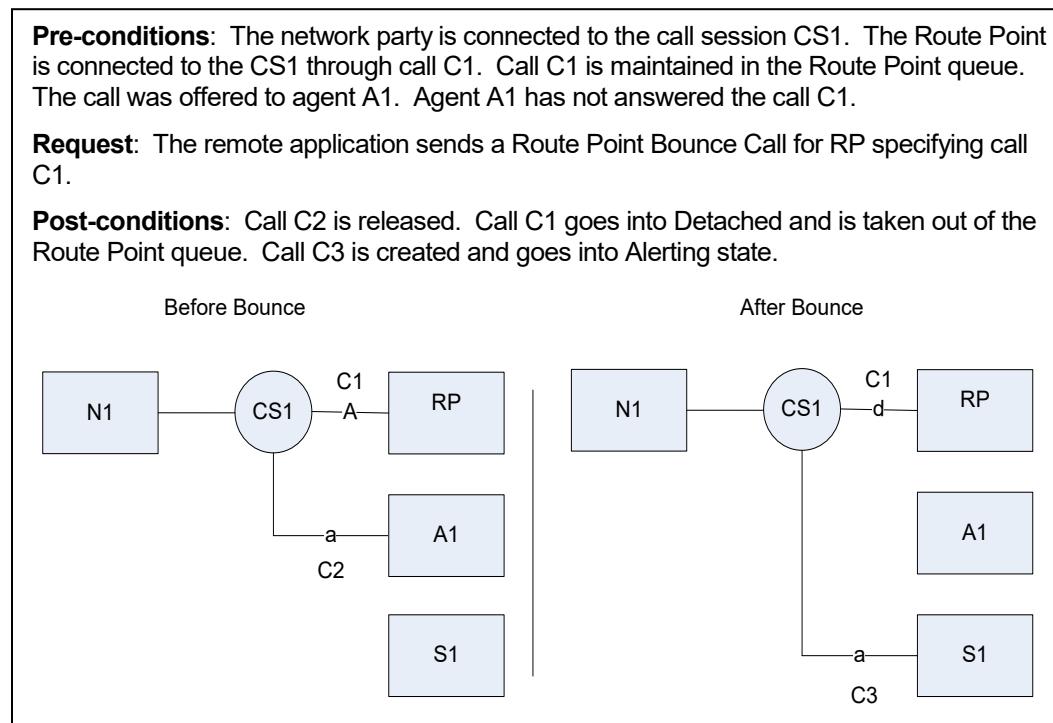


Figure 92 Bounced Call is Transferred to New Destination State Changes

5.4.9.2 Request

A remote application can issue this request by sending a `RoutePointBounceCallRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>routepointId</code>	SubscriberId	Yes	The subscriber ID of the route point.
<code>callId</code>	CallId	Yes	The call ID of the call to bounce.

5.4.9.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.4.9.4 Associated Events

In case 1, when a Route Point Bounce Call request is accepted the following events are generated:

- A *Route Point Call Bounced* event against the Route Point's call (C1 in the diagram).
- A *Call Released* event against the call that was offered to the agent (C2 in the diagram).

In case 2, when a Route Point Bounce Call request is accepted the following events are generated:

- A *Route Point Call Bounced* event against the Route Point's call (C1 in the diagram). The event specifies that the call was transferred.
- A *Call Released* event against the call that was offered to the agent (C2 in the diagram).
- A *Call Received* event against the transferred-to subscriber's call (C3 in the diagram).

5.4.9.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110407	User is not a route point.
400	101001	State for call <callId> is not valid.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.4.10 Blind Transfer

5.4.10.1 Description

This request is used to transfer a call queued at a Route Point to a new destination.

The following diagram provides information on the state and topology changes involved.

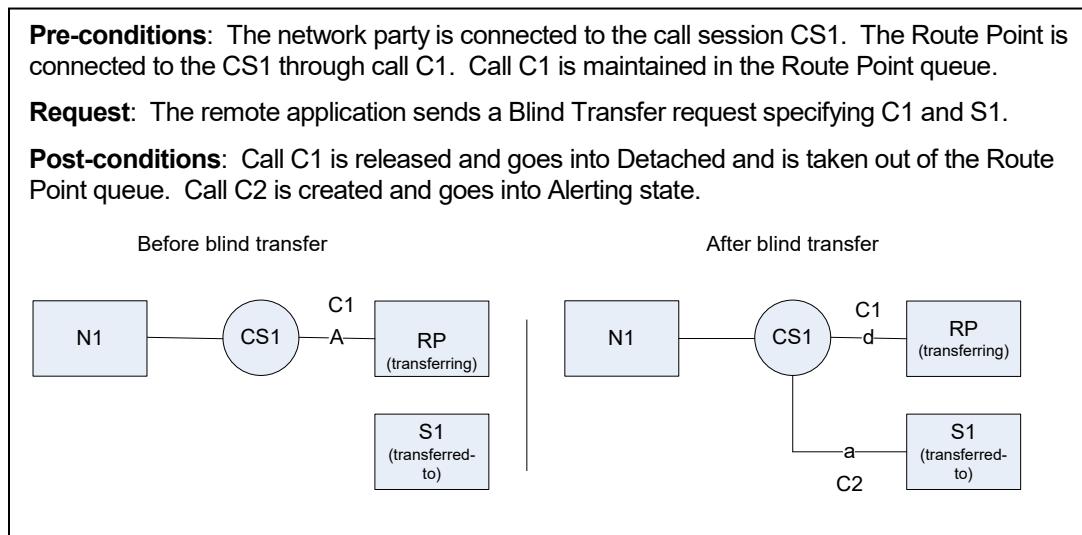


Figure 93 Blind Transfer State Changes

5.4.10.2 Request

A remote application can issue this request by sending a `RoutePointBlindTransferRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>routepointId</code>	SubscriberId	Yes	The subscriber ID of the route point.
<code>callId</code>	CallId	Yes	The call ID of the call to transfer.
<code>address</code>	Address	Yes	The destination address for the transfer.

5.4.10.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.4.10.4 Associated Events

When a Route Point Blind Transfer request is accepted, the following events are generated:

- A *Route Point Call Transferred* event against the Route Point's call (C1 in the diagram).
- A *Call Received* event against the transferred-to subscriber's call (C2 in the diagram).

If the calling subscriber is not available (for example, busy) then the following events are generated:

- A *Route Point Call Transferred* event against the Route Point's call (C1 in the diagram).



- A *Call Released* event with a ReleaseCause set to “busy” against the transferred-to subscriber’s call (C2 in the diagram).

5.4.10.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110407	User is not a route point.
400	110402	Route Point failed.
400	101001	State for call <callId> is not valid.
400	100007	Blocked by Outgoing Calling Plan service.
400	100009	Blocked by Communication Barring service.
400	101005	Diversions are inhibited for call <callId>.
400	101000	Call <callId> has no remote call.
400	100010	Blocked by translations.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.4.11 Release Call

5.4.11.1 Description

This request is used by a remote application to release a call maintained in a Route Point queue. As a result of this request, the call is removed from the Route Point queue.

If the call session was offered to an agent, the associated call is also released. However, if the call was offered and answered by an agent, the call cannot be released by this command, as the call is no longer in the Route Point queue.

The following two diagrams provide information on the state and topology changes involved.

5.4.11.1.1 Case 1: Release Queued Call

Pre-conditions: The network party is connected to the call session CS1. The Route Point is connected to CS1 through call C1. Call C1 is maintained in the Route Point queue.

Request: The remote application sends a Route Point Release Call request for RP specifying call C1.

Post-conditions: Call C1 is destroyed as well as the Call Session CS1.

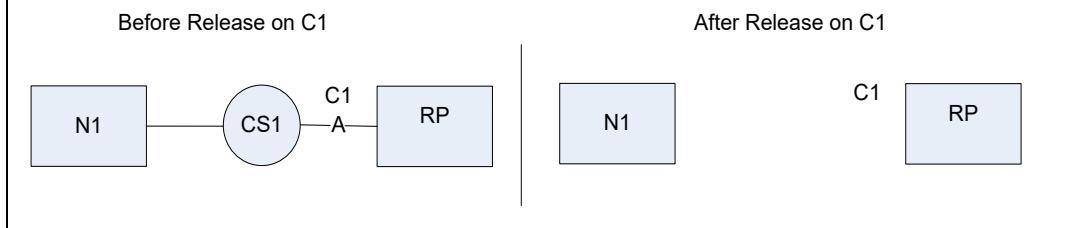


Figure 94 Release Queued Call

5.4.11.1.2 Case 2: Release Offered Call

Pre-conditions: The network party is connected to the call session CS1. The Route Point is connected to CS1 through call C1. Call C1 is maintained in the Route Point queue. Call C2 is offered to agent A1.

Request: The remote application sends a Route Point Release Call for RP specifying call C1.

Post-conditions: Call C1 and C2 are destroyed as well as the Call Session CS1.

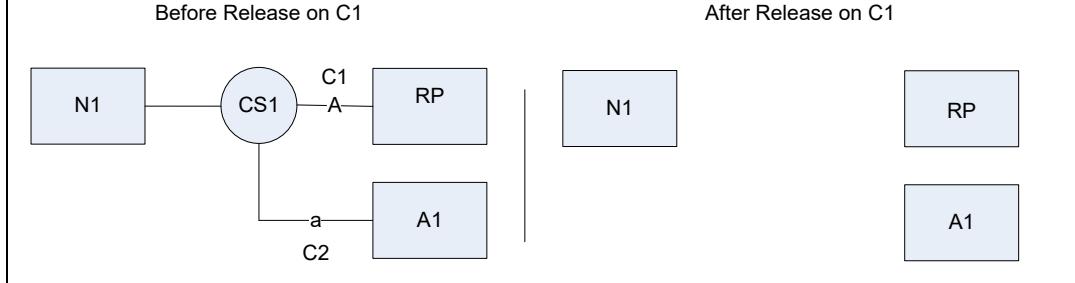


Figure 95 Release Offered Call

5.4.11.2 Request

A remote application can issue this request by sending a `RoutePointReleaseCallRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>routepointId</code>	SubscriberId	Yes	The subscriber ID of the route point.
<code>callId</code>	CallId	Yes	The call ID of the call to release.



5.4.11.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.4.11.4 Associated Events

In case 1, when the Route Point Release Call request is accepted, a *Route Point Call Released* event is generated against the Route Point's call (C1 in the diagram).

In case 2, when the Route Point Release Call request is accepted, a *Route Point Call Released* event is generated against the Route Point's call (C1 in the diagram). A Call Released event is generated against the agent's call (C2 in the diagram).

5.4.11.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110407	User is not a route point.
400	110402	Route Point failed.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.4.12 Fail Call

5.4.12.1 Description

This request is used to mark a call queued at a Route Point has failed resulting in running the failed call policy on the call. As a result, the call is sent to the configured failed destination immediately.

A call cannot be failed if the call is in the Route Point queue but is currently being offered to an agent.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: The network party is connected to the call session CS1. The Route Point is connected to the CS1 through call C1. Call C1 is maintained in the Route Point queue. Subscriber S1 is configured as the failed destination for Route Point RP. Typically, S1 can be another RP or ACD.

Request: The remote application sends a Route Point Fail Call request for RP specifying call C1.

Post-conditions: Call C1 is released and call C2 is transferred to the failed destination.

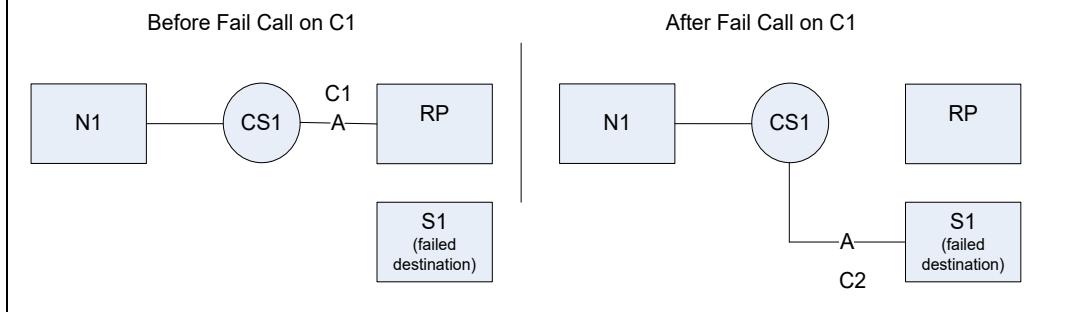


Figure 96 Fail Call State Changes

5.4.12.2 Request

A remote application can issue this request by sending a `RoutePointFailCallRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>routepointId</code>	SubscriberId	Yes	The subscriber ID of the route point.
<code>callId</code>	CallId	Yes	The call ID of the call to fail.

5.4.12.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.4.12.4 Associated Events

In case 1, when a Route Point Fail Call request is accepted the following events are generated:

- A *Route Point Call Failed* event against the Route Point's call (C1 in the diagram).
- A *Call Received* event against the failed destination (S1 the diagram).

5.4.12.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110407	User is not a route point.
400	110402	Route Point failed.
400	110401	An outgoing call for a route point cannot be failed.

statusCode	errorCode	Summary
400	101001	State for call <callId> is not valid.
400	110408	Per call failure policy is disabled.
400	101005	Diversions are inhibited for call <callId>.
For <i>System</i> errors, see section 5.1.3.5 System Errors .		
For <i>Target User</i> errors, see section 5.1.3.1 Target User Errors .		
For <i>Call ID</i> errors, see section 5.1.3.3 CallId Errors .		

5.4.13 Get Route Point State

5.4.13.1 Description

This request is used to get the Route Point state. If the request is accepted, then the returned response contains the state information.

5.4.13.2 Request

A remote application can issue this request by sending a GetRoutePointStateRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>routepointId</i>	SubscriberId	Yes	The subscriber ID of the route point.

5.4.13.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>routePointState</i>	RoutePointState	Yes	Contains the state of the route point. Only present when a successful response is returned (that is, statusCode is set to "200").

5.4.13.4 Associated Events

There is no event associated with this request.

5.4.13.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110407	User is not a route point.
For <i>System</i> errors, see section 5.1.3.5 System Errors .		
For <i>Target User</i> errors, see section 5.1.3.1 Target User Errors .		

5.4.14 Modify Route Point State

5.4.14.1 Description

This request is used to put an entire Route Point in the *Failed* state or *Normal* (Recovered) state.

5.4.14.1.1 Transition to Normal State

If the Route Point is put in the *Normal* state, and if the request is accepted, then all new calls received are no longer forwarded to the failed destination. Calls that were previously sent to the failed destination are not returned to the Route Point.

Transition to the *Normal* state does not involve any call state or topology changes.

5.4.14.1.2 Transition to Failed State

If the Route Point is put in the *Failed* state, and if the request is accepted, then the Route Point failure policy is applied causing all calls queued in the Route Point to be sent to the configured failed destination. The calls being delivered to the failure destination are throttled by Cisco BroadWorks to not impact performance on the system.

This does not apply to calls being offered to agents. These calls continue to be offered until bounced which then cause them to be routed to the failed destination with all other queued calls. All new calls received are forwarded to the failed destination.

Note that calls that were answered by an agent and that are therefore in the *Detached* state at the Route Point are maintained at the Route Point.

The state and topology changes are described by the following two cases.

Case 1: Call in queue

In this case, a queued call (not offered to an agent) is failed. The following diagram provides information on the state and topology changes involved.

Pre-conditions: The network party is connected to the call session CS1. The Route Point is connected to the CS1 through call C1. Call C1 is maintained in the Route Point queue. S1 is configured as the failed destination for Route Point RP.

Request: The remote application sends a Route Point Fail request.

Post-conditions: Call C1 is released and call C2 is transferred to the failed destination.

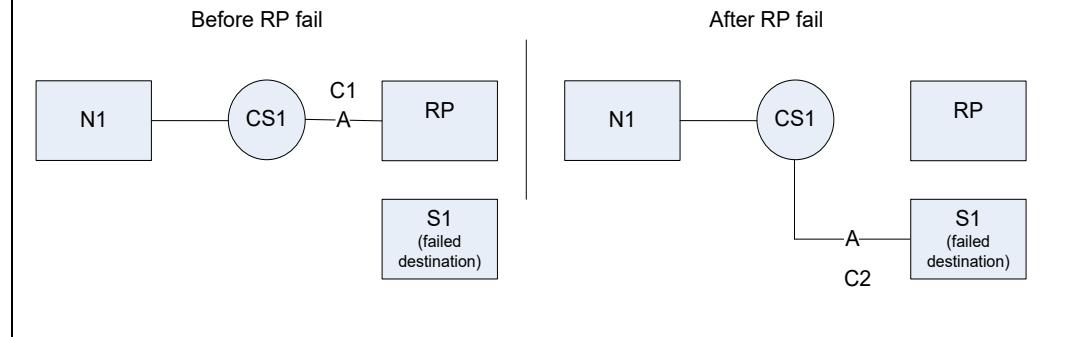


Figure 97 Call in Queue

Case 2: Call in queue but offered to an agent

In this case, a queued call offered to an agent is not failed.

The following diagram provides information on the state and topology changes involved.

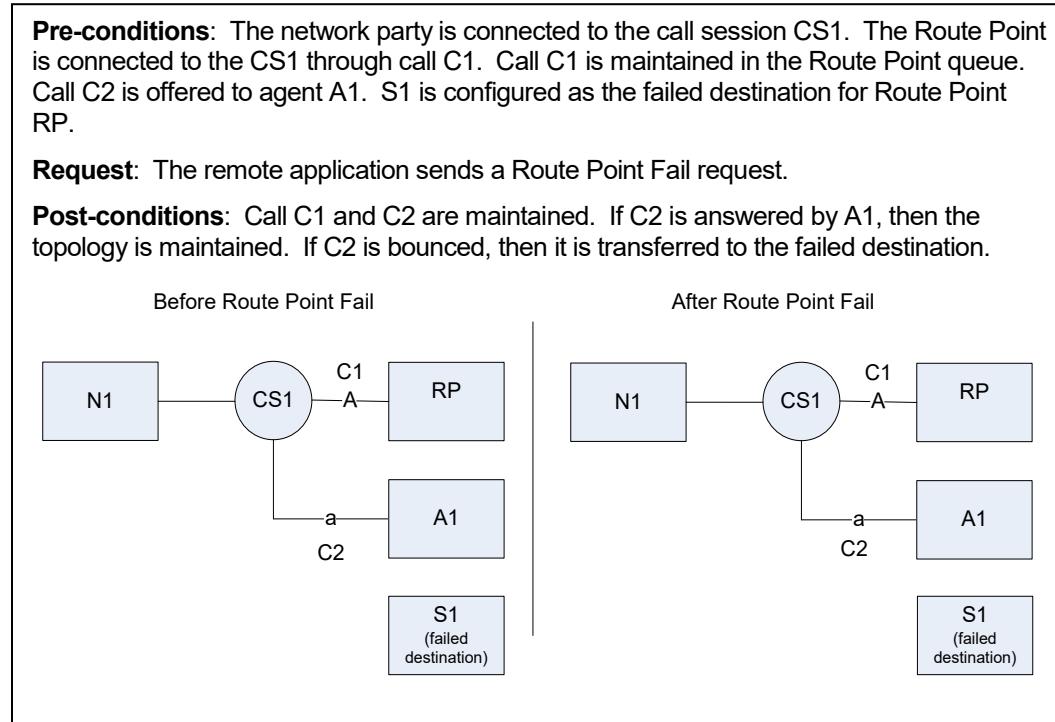


Figure 98 Call in Queue but Offered to Agent

5.4.14.2 Request

A remote application can issue this request by sending a `ModifyRoutePointStateRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>routepointId</code>	SubscriberId	Yes	The subscriber ID of the route point.
<code>routePointState</code>	RoutePointState	Yes	The new state of the route point.

5.4.14.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.4.14.4 Associated Events

5.4.14.4.1 Transition to Normal State

A *Route Point Recovered* event is generated against the Route Point when the Route Point reaches the *Normal* state.

5.4.14.4.2 Transition to Failed State

In case 1, when a Route Point Fail Call request is accepted the following events are generated:

- A *Route Point Failed* event against the Route Point.
- A *Route Point Call Failed* event against the Route Point's call (C1 in the diagram).
- An *ACD Call Added* event against the failed destination (ACD in the diagram).

In case 2, when a Route Point Fail Call request is accepted the following events are generated:

- A *Route Point Failed* event against the Route Point.

5.4.14.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110407	User is not a route point.
400	110403	Failover policy is disabled.
400	110410	Failover in progress.
400	110412	External system is not ready.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.5 ACD Requests (AS only)

5.5.1 Get ACD

5.5.1.1 Description

This request is used to get the list of calls maintained in an ACD queue.

5.5.1.2 Request

A remote application can issue this request by sending a GetACDRequest. This request contains the common parameters listed in section [5.1.1 Request](#).

Parameter	Parameter Type	Required	Description
callcenterId	SubscriberId	Yes	The subscriber ID of the ACD.

5.5.1.3 Response

The response contains the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
ACDQueue	ACDQueue	No	The detail of the ACD queue. Only present if the statusCode was set to "200".

5.5.1.4 Associated Events

No events are associated with this request

5.5.1.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110203	User is not a call center.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.5.2 Blind Transfer

5.5.2.1 Description

This request is used to transfer a call queued at an ACD to a new destination.

The following diagram provides information on the state and topology changes involved.

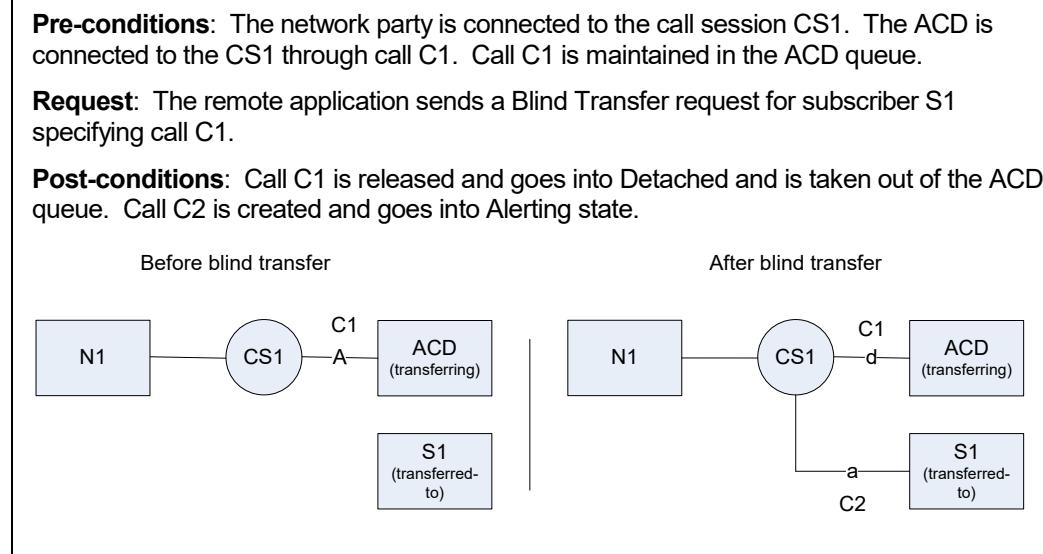


Figure 99 Blind Transfer State Changes

5.5.2.2 Request

A remote application can issue this request by sending an ACDBlindTransferRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
callcenterId	SubscriberId	Yes	The subscriber ID of the ACD.
callId	CallId	Yes	The call ID of the call to transfer.
phonenos	Address	Yes	The destination address for the transfer.

5.5.2.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.5.2.4 Associated Events

When an ACD Blind Transfer request is accepted, the following events are generated:

- An *ACD Call Transferred* event against the ACD's call (C1 in the diagram).
 - A *Call Received* event against the transferred-to subscriber's call (C2 in the diagram).
- If the calling user is not available (for example, busy) then the following events are generated:
- An *ACD Call Transferred* event against the ACD's call (C1 in the diagram).
 - A *Call Released* event with a ReleaseCause set to "busy" against the transferred-to subscriber's call (C2 in the diagram).

5.5.2.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110203	User is not a call center.
400	110206	Cannot perform function – mandatory entrance message is playing.
400	101001	State for call <callId> is not valid.
400	101000	Call <callId> has no remote call.
400	110541	Call <callId> is an answered Push-To-Talk call.
400	100008	Remote party for call <callId> is a virtual subscriber.
400	100017	Blocked by maximum concurrent redirections policy.
400	100010	Blocked by translations.
400	100018	Blocked by Intercept Group or Intercept User service.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Address errors, see section 5.1.3.2 Address Error .		

5.5.3 Promote Call

5.5.3.1 Description

This request is used to change the priority of a call queued in an ACD. The call can only be promoted not demoted. For information on how the ACDPriority value can only be reduced, see section [8 Type Definitions](#). Cisco BroadWorks rejects any request to move a call to a lower priority (that is, increase the ACDPriority value). Only calls that have not been bounced and that are not being played a mandatory entrance message can be promoted.

This request does not involve any call state or topology changes.

5.5.3.2 Request

A remote application can issue this request by sending an ACDPromoteCallRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>callcenterId</i>	SubscriberId	Yes	The subscriber ID of the ACD.
<i>callId</i>	CallId	Yes	The call ID of the call to promote.
<i>ACDPromote</i>	ACDPromote	Yes	This parameter carries the priority to promote the call.

5.5.3.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>queueCallOrder</i>	QueueCallOrder	No	The new position of the promoted call. Only present when a successful response is returned (that is, statusCode is set to "200").

5.5.3.4 Associated Events

When an ACDPromote request is accepted, an ACDCallPromoted event is issued against the ACD.

5.5.3.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110203	User is not a call center.
400	110209	Only calls in a premium call center can be promoted.
400	110205	Cannot perform function – call is bounced.
400	110210	Calls can only be promoted to a higher priority.
400	101001	State for call <callId> is not valid.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.5.4 Reorder Call

5.5.4.1 Description

This request is used to move the call in an ACD queue to the specified position.

The position specifies the location of a call in an ACD queue. The call at position 1 in the queue is the next call to be distributed to an agent. A remote application can modify the position of a call if its priority is set to "0-Highest", if it has not been bounced and if it is not being played a mandatory entrance message.

The new position is returned in the response. Note that the new position may not always match the requested position due to the dynamic nature of the queue.

This request does not involve any call state or topology changes.

5.5.4.2 Request

A remote application can issue this request by sending an ACDReorderCallRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>callcenterId</i>	SubscriberId	Yes	The subscriber ID of the ACD.
<i>callId</i>	CallId	Yes	The call ID of the call to promote.
<i>index</i>	PositiveInteger	Yes	The new position for the call's entry in the ACD queue.

5.5.4.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>queueCallOrder</i>	QueueCallOrder	No	The new position of the promoted call. Only present when a successful response is returned (that is, statusCode is set to "200").

5.5.4.4 Associated Events

When an ACD Reorder request is accepted, an *ACD Call Reordered* event is issued against the ACD.

Note that this position change does impact the position of other calls. No event is issued to notify the new position for these affected calls.

5.5.4.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110203	User is not a call center.
400	110205	Cannot perform function – call is bounced.
400	110207	Cannot perform function – only calls in the highest priority can be reordered.

statusCode	errorCode	Summary
400	101001	State for call <callId> is not valid.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.5.5 Release Call

5.5.5.1 Description

This request is used by a remote application to release a call maintained in an ACD queue. As a result of this request, the call is removed from the ACD queue.

If the call session was offered to an agent, the associated call is also released. However, if the call was offered and answered by an agent, the call cannot be released by this command, as the call is no longer in the Route Point queue.

The following two diagrams provide information on the state and topology changes involved.

5.5.5.1.1 Case 1: Release Queued Call

Pre-conditions: The network party is connected to the call session CS1. The Route Point is connected to CS1 through call C1. Call C1 is maintained in the ACD queue.

Request: The remote application sends an ACD Release Call request for the ACD specifying call C1.

Post-conditions: Call C1 is released as well as the Call Session CS1.

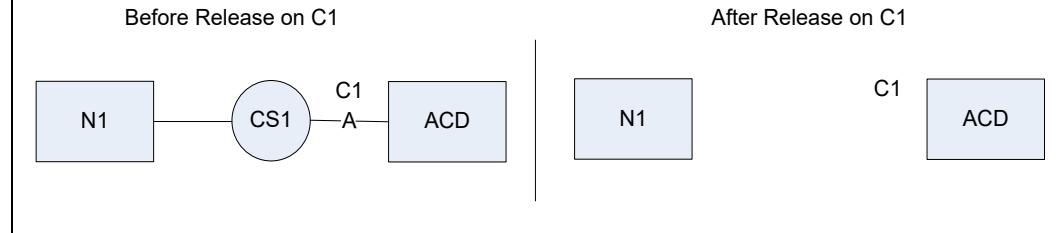


Figure 100 Release Queued Call

5.5.5.1.2 Case 2: Release Offered Call

Pre-conditions: The network party is connected to the call session CS1. The ACD is connected to CS1 through call C1. Call C1 is maintained in the ACD queue. Call C2 is offered to agent A1.

Request: The remote application sends a ACD Release Call for the ACD specifying call C1.

Post-conditions: Call C1 and C2 are released as well as the Call Session CS1.

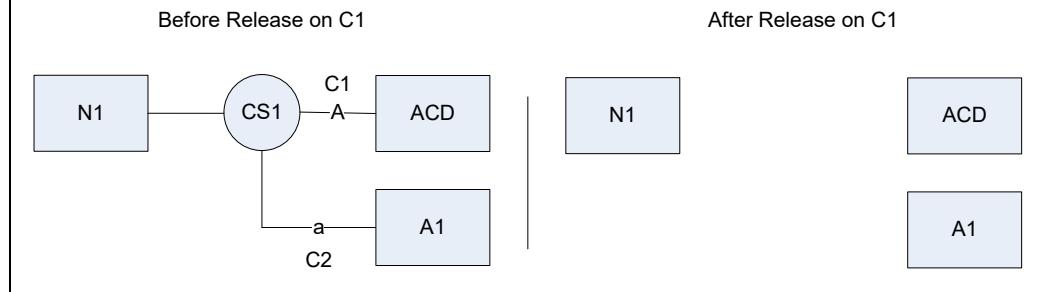


Figure 101 Release Offered Call

5.5.5.2 Request

A remote application can issue this request by sending an `ACDReleaseCallRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>callcenterId</code>	SubscriberId	Yes	The subscriber ID of the ACD.
<code>callId</code>	CallId	Yes	The call ID of the call to release.

5.5.5.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.5.5.4 Associated Events

In case 1, when the ACD Release Call request is accepted, an *ACD Call Released* event is generated against the ACD's call (C1 in the diagram).

In case 2, when the ACD Release Call request is accepted, an *ACD Call Released* event is generated against the ACD's call (C1 in the diagram). A *Call Released* event is generated against the agent's call (C2 in the diagram).

5.5.5.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110203	User is not a call center.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

statusCode	errorCode	Summary
------------	-----------	---------

For Call ID errors, see section [5.1.3.3 CallId Errors](#).

5.6 Agent Requests (AS only)

5.6.1 Get Agent State

5.6.1.1 Description

This request is used to get the agent state. If the service user ID of the assigned call centers does not have a domain part, the domain must be considered equal to the system default domain.

5.6.1.2 Request

A remote application can issue this request by sending a GetAgentStateRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the agent.

5.6.1.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>callCenter</i>	CallCenter	Yes	Contains the agent state information along with the list of assigned call centers with the current join status for each call center. For more information, see section 8.10.8 Call Center .

5.6.1.4 Associated Events

There are no events associated to this request.

5.6.1.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
401	4008	User not found: <userId>.
401	4003	Unauthorized request: <request>.

For System errors, see section [5.1.3.5 System Errors](#).

5.6.2 Set Agent State

5.6.2.1 Description

This request is used to set the agent state. Cisco BroadWorks performs basic validation on the request and if it is accepted, change the agent state and returns a successful response. All agent state transitions are accepted, as explained in section [4.4.5 Agent](#).

5.6.2.2 Request

A remote application can issue this request by sending a SetAgentStateRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the agent.
<i>callCenter</i>	CallCenter	Yes	The associated agent state information. For more information on callCenter type, see section 8.10.8 Call Center .

5.6.2.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.6.2.4 Associated Events

When a Set Agent State request is accepted, an *Agent State* event is generated indicating that the agent state has been changed.

5.6.2.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
401	4008	User not found: <userId>.
401	4003	Unauthorized request: <request>.

For System errors, see section [5.1.3.5 System Errors](#).

5.6.3 Tag Disposition Code

5.6.3.1 Description

This request is used to set a disposition code as explained in section [4.4.5.2 Disposition Code](#). Cisco BroadWorks performs basic validation on the request and if it is accepted, set the disposition code with to the last treated call by the agent. The agent must be in the *Wrap-up* state.

5.6.3.2 Request

A remote application can issue this request by sending a TagDispositionCodeRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the agent.
<i>dispositionCode</i>	AgentDispositionCode	Yes	The disposition code.

5.6.3.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.6.3.4 Associated Events

When an Add Disposition Code request is accepted, an *Agent Disposition Code Added* event is generated indicating that a code was added by the agent against a call.

5.6.3.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110202	User is not assigned the Call Center-Premium service.
400	110106	Agent must be in wrap-up state when call is not specified.
400	110109	Disposition codes disabled for this call center.
400	110107	Disposition code is not valid for this call center.
400	110102	Inactive disposition code.
400	110100	Cannot apply a disposition code to call <callId> since it is not a call center call.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

For Call ID errors, see section [5.1.3.3 CallId Errors](#).

5.6.4 Tag Disposition Code to Call ID

5.6.4.1 Description

This request is used to set a disposition code as explained in section [4.4.5.2 Disposition Code](#). Cisco BroadWorks performs basic validation on the request and, if it is accepted, sets the disposition code with the associated call ID for the agent. The call must be an ACD call in the *Active*, *Held*, or *Remote Held* state.

5.6.4.2 Request

A remote application can issue this request by sending a TagDispositionCodeRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the agent.
<i>callId</i>	CallId	No	The associated call ID. Required if the agent is not in <i>Wrap-up</i> state.
<i>dispositionCode</i>	AgentDispositionCode	Yes	The disposition code.

5.6.4.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.6.4.4 Associated Events

When an Add Disposition Code request is accepted, an *Agent Disposition Code Added* event is generated indicating that a code was added by the agent against a call.

5.6.4.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110202	User is not assigned the Call Center-Premium service.
400	110106	Agent must be in <i>Wrap-up</i> state when call is not specified.
400	110109	Disposition codes disabled for this call center.
400	110107	Disposition code is not valid for this call center.
400	110102	Inactive disposition code.
400	110100	Cannot apply a disposition code to call <callId> since it is not a call center call.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

For Call ID errors, see section [5.1.3.3 CallId Errors](#).

5.6.5 Escalate to Supervisor

5.6.5.1 Description

This request is used to originate a call session from the specified agent to an available supervisor. When escalating a call to the supervisor, the existing call handled by the agent is placed on hold.

The request can optionally specify the supervisor's address. If the request does not specify the address of the supervisor then Cisco BroadWorks determines an available supervisor from the list of supervisors assigned to that particular agent.

Two conditions must be met for the call to go to an available supervisor.

- The supervisor should not be on a call.
- The supervisor should not have DND set.

An error is returned by Cisco BroadWorks if no supervisor is found.

If the supervisor is available, then Cisco BroadWorks initiates a call towards the agent, assigns a call ID and an external tracking ID to the connection, and returns it in the response.

As for the Dial request, Cisco BroadWorks alerts the agent's endpoint prompting the subscriber to go off-hook. In some cases, the endpoint can support protocol extensions to immediately go off-hook without requiring a subscriber action.

Once the agent's endpoint has gone off-hook, the call to the supervisor is resumed. The following diagram provides information on the state and topology changes involved.

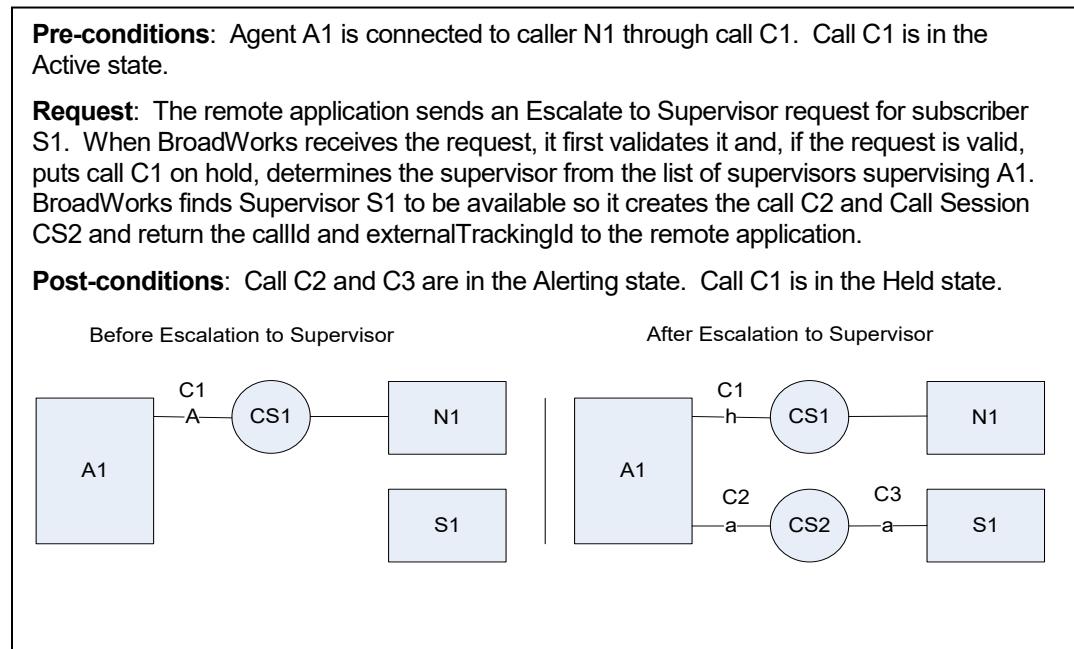


Figure 102 Escalate to Supervisor State Changes

5.6.5.2 Request

A remote application can issue this request by sending an EscalateToSupervisorRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
subscriberId	SubscriberId	Yes	The subscriber ID of the calling user.
address	Address	No	The address of the supervisor to whom the call is being escalated. If not present, Cisco BroadWorks determines an available supervisor.

Parameter	Parameter Type	Required	Description
<i>location</i>	ClickToDialLocationEnum	No	When present, specifies the location of the calling user to ring for the call. When not present, locations are alerted per the calling user's profile on Cisco BroadWorks.
<i>locationAddress</i>	NonEmptyToken	No	Specific address of location to ring. This element should only be present if the "location" element is present and set to "BroadWorksAnywhere" or "SharedCallAppearance". When the "location" element is set to "BroadWorksAnywhere", the locationAddress is optional and contains a directory number. The directory number can be in E.164 format or it can just contain digits. When not present, all BroadWorks Anywhere locations are alerted. The following are valid directory numbers for the locationAddress element: <ul style="list-style-type: none">▪ 9725551000▪ +19725551000 When the "location" element is set to "SharedCallAppearance", the locationAddress is optional and contains a Cisco BroadWorks address of record. The address of record may be user@domain for SIP locations or user for MGCP locations. When not present, all Shared Call Appearance locations are alerted. The following are valid address of records for the locationAddress element: <ul style="list-style-type: none">▪ 9725551000@broadsoft.com▪ 9725551000

5.6.5.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
callStartInfo	CallStartInfo	No	Contains the call ID and external Tracking ID of the new call. It is only present when the statusCode is set to "201".

5.6.5.4 Associated Events

The same events as for a Dial request are produced.

Additionally, if no supervisor is found available or if the address specified in the request does not correspond to an available subscriber, then a *Call Releasing* event is generated with the *releaseCause* set while the agent is connected to a media treatment. A *Call Released* event is finally generated once the treatment stops playing or if the subscriber hangs up. The event also contains the same *releaseCause* value.

5.6.5.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111000	Invalid location value <location>.
400	111001	Cannot find location <locationAddress>.
400	111002	The locationAddress not allowed for location <location>.
400	111003	User is not assigned the <service name> service or there is no location assigned to this service.
400	111004	User does not have a primary endpoint.
400	110820	Blocked by Remote Office.
400	110201	User is not assigned the either the Call Center-Standard or Call Center-Premium service.
400	100006	Endpoint state is not valid.
400	110581	Emergency call is present.
400	110582	Blocked by prepaid service.
400	110580	User has no charge address.
400	110646	Not allowed to have more than one call outside the conference.
400	100019	Blocked by maximum simultaneous calls policy.
400	110011	Answer confirmation in progress.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Address errors, see section 5.1.3.2 Address Error .		
For Address Destination errors, see section 5.1.3.4 Address Destination Errors .		

5.6.6 Emergency Call to Supervisor

5.6.6.1 Description

The Emergency Call to Supervisor request is used by an agent to immediately contact a supervisor while the agent is handling a call. The existing agent call is “not” placed on ‘hold’ when an emergency call is initiated. Instead, the supervisor is immediately conferenced into the call.

This request requires that the agent has the Three-Way Calling or N-Way Calling service and a Call Center service (basic, standard or premium).

The request may optionally specify the supervisor’s address. If the request does not specify the address of the supervisor then Cisco BroadWorks determines an available supervisor from the list of supervisors assigned to that particular agent.

Two conditions must be met for the call to go to an available supervisor:

- The supervisor should not be on a call.
- The supervisor should not have DND set.

If no supervisor is found, an error is returned by Cisco BroadWorks.

If the supervisor is available, then Cisco BroadWorks initiates a call from the agent towards the supervisor, assigns a call ID and an external tracking ID to the connection. The supervisor endpoint is then alerted. Once the supervisor has answered the call, Cisco BroadWorks creates a three-way conference between the supervisor, the agent, and the caller. This establishes an emergency escalation between the three parties. The agent is the controller for this conference.

If the warning tone option is enabled, the agent is given the emergency escalation warning tone prior to the conference creation.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: Call C1 can be in the Held or Active state.

Request: The remote application sends a Emergency Call to Supervisor request for A. When BroadWorks receives the request, it first validates it and, if the request is valid determines the supervisor from the list of available supervisors. BroadWorks finds Supervisor S1 to be available so it creates the call C2 and C3 and Call Session CS2 and returns the callId and externalTrackingId to the remote application.

Post-conditions: Supervisor S1 endpoint is alerted. When supervisor S1 answers the call, a conference is created with A1 as the controller and S1 and N as participants. C1 remains in the same state as initially. C2 goes into Active state.

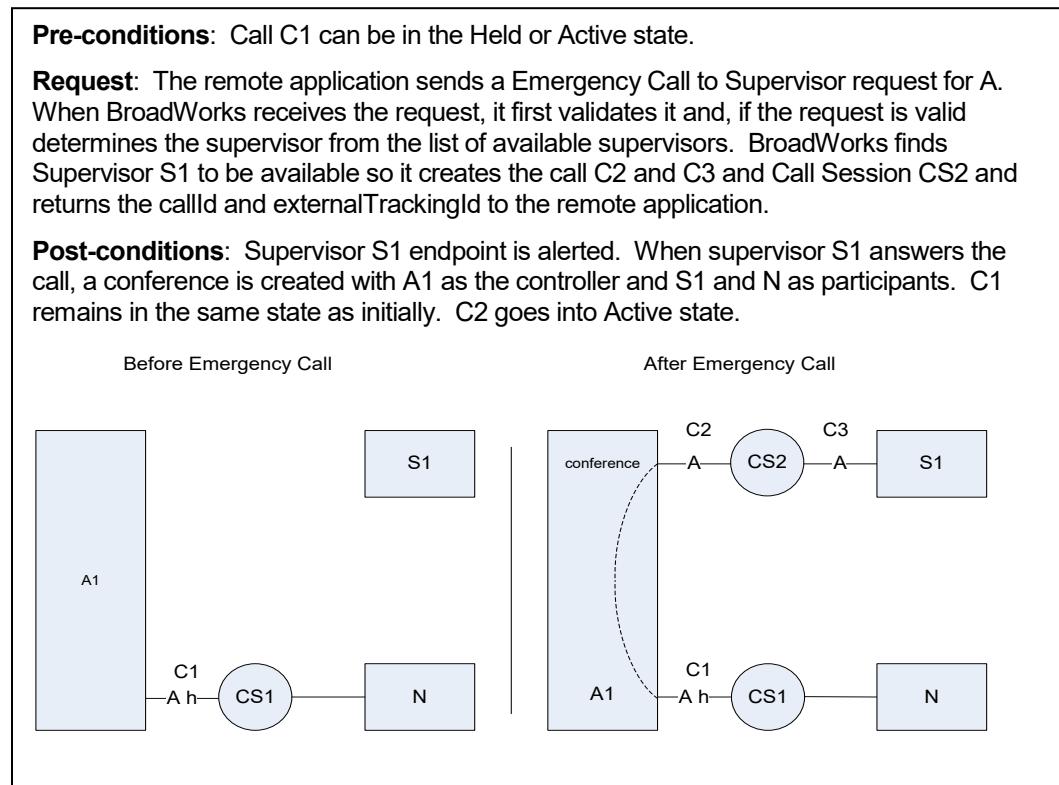


Figure 103 Emergency Call to Supervisor State Changes

5.6.6.2 Request

A remote application can issue this request by sending an `EmergencyCallToSupervisorRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID of the agent.
<code>address</code>	Address	No	The address of the supervisor. If not present, Cisco BroadWorks determines an available supervisor.
<code>callId</code>	CallId	Yes	The call ID for which the emergency call applies.

5.6.6.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>callStartInfo</code>	CallStartInfo	No	Contains the call ID and external tracking ID of the new call. It is only present when the statusCode is set to "201".

5.6.6.4 Associated Events

The following time sequence diagram illustrates the events produced when an *Emergency Call to Supervisor* request is accepted.

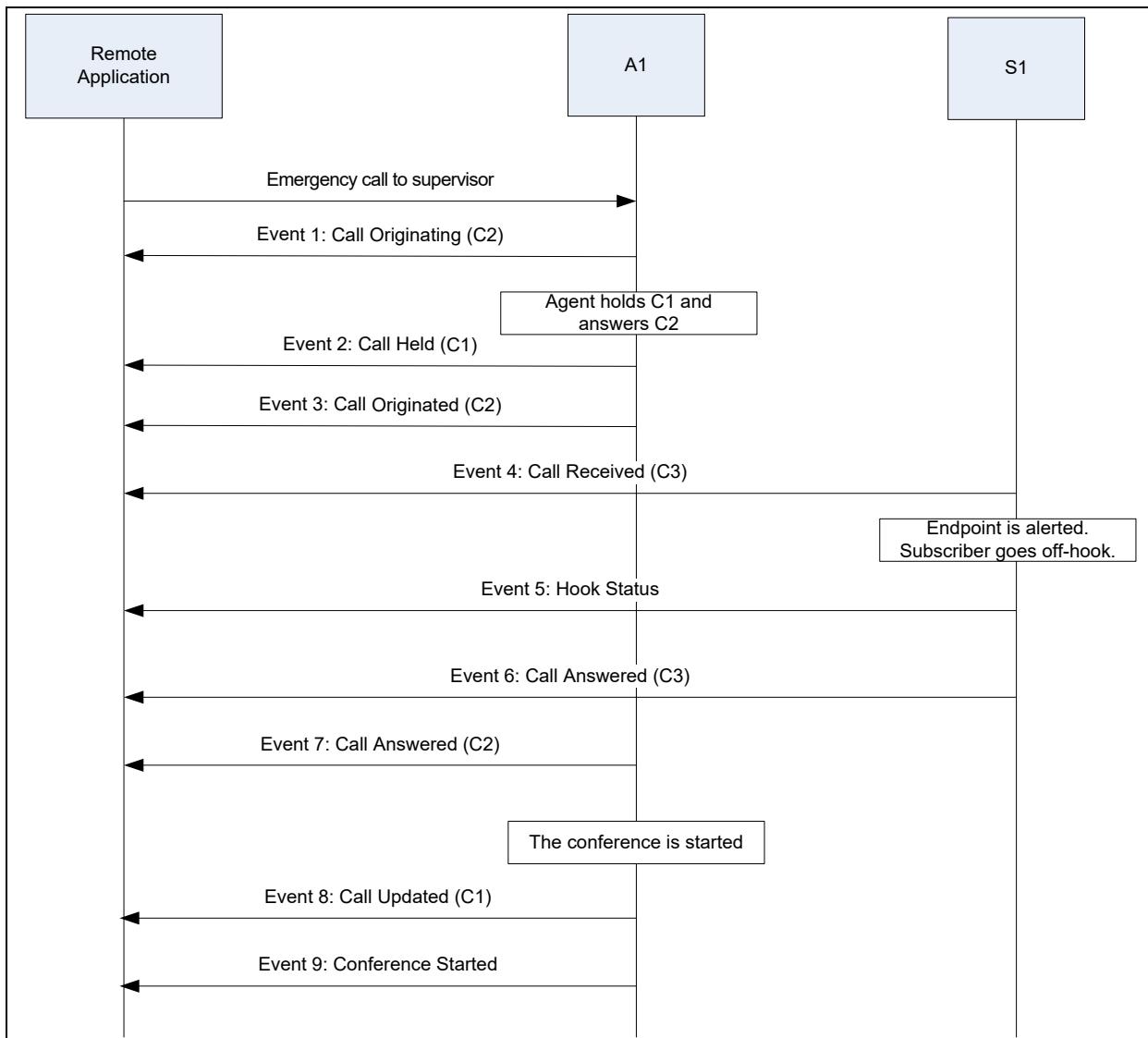


Figure 104 Emergency Call to Supervisor Events

5.6.6.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110200	User is not assigned the Call Center-Basic, Call Center-Standard, or Call Center-Premium service.
400	110650	User is not assigned the Three-Way Calling or N-Way Calling service.
400	101002	Call <callId> not found.
400	110103	User <userId> is not a supervisor for this agent.
400	110104	All supervisors for this agent are busy or cannot receive an emergency call.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Address errors, see section 5.1.3.2 Address Error .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.7 Meet-Me Conference Requests (AS only)

5.7.1 Get Meet-Me Conference

5.7.1.1 Description

This request is used to get details of a Meet-Me conference. The information provided contains details regarding the conference itself as well as the list of participants who joined the conference.

5.7.1.2 Request

A remote application can issue this request by sending a GetMeetMeConferenceRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>bridgeld</i>	SubscriberId	Yes	The subscriber ID of the Meet-Me conference bridge.
<i>conferenceId</i>	MeetMeConferenceId	Yes	The conference ID

5.7.1.3 Response

The response contains the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>MeetMeConference</i>	MeetMeConference	No	The detail of the Meet-Me conference. Only present if the statusCode was set to "200".



5.7.1.4 Associated Events

No events are associated with this request.

5.7.1.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110888	The target user is not a Meet-Me virtual subscriber.
400	110873	Invalid Conference ID.
400	110889	The conference has not started.
400	110893	The server hosting the conference is not available.

For *System* errors, see section [5.1.3.5 System Errors](#).

For *Target User* errors, see section [5.1.3.1 Target User Errors](#).

5.7.2 Outgoing Dial

5.7.2.1 Description

This request is used to originate a call session from the Meet-Me conference to the specified destination. If the request is accepted, then Cisco BroadWorks initiates a call to the target destination.

The target is prompted to join the conference once the call is answered. The call is considered to have joined the conference only once the target has confirmed acceptance. The target does not appear to have a participant until then.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: None.

Request: The remote application sends an Outgoing Dial request on behalf of the Meet-Me conference with the destination phone number of S1.

Post-conditions: Call Session CS1 and call C1 and C2 are created. Both calls are in the Alerting state. S1 is not considered to have joined the conference until the call is answered and S1 has confirmed to join the conference.

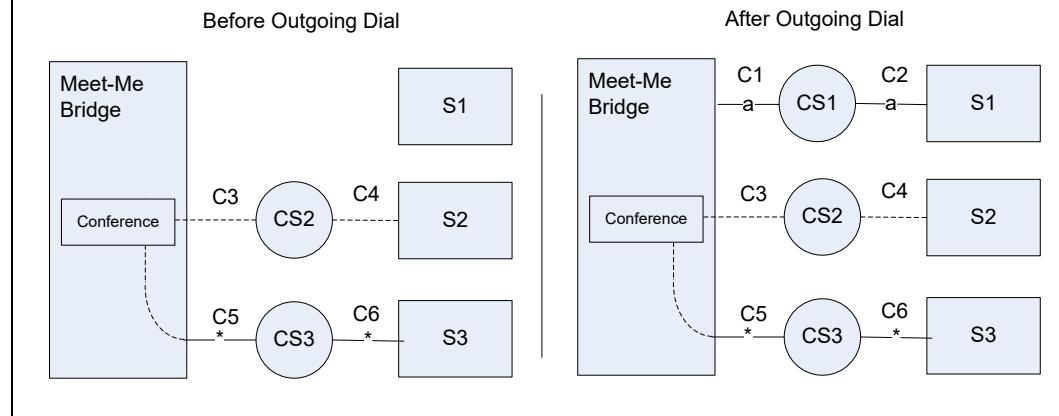


Figure 105 Meet-Me Outgoing Dial

5.7.2.2 Request

A remote application can issue this request by sending a `MeetMeConferenceOutgoingDialRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>bridgeId</code>	SubscriberId	Yes	The subscriber ID of the Meet-Me conference bridge.
<code>conferenceId</code>	MeetMeConferenceId	Yes	The conference ID or the moderator PIN. If the moderator PIN is used, the resulting call leg inherits the moderator's functions since creation was triggered using the moderator PIN. Moderator functions are then available to the answering user of that call leg.
<code>address</code>	Address	Yes	The address to call.
<code>uniqueIdentifier</code>	String	No	Allows a participant to identify themselves with a code when joining the conference. This parameter should contain digits (0 to 9). A maximum of 10 digits can be specified.

5.7.2.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.7.2.4 Associated Events

The following time sequence diagram illustrates the event produced when a *Meet-Me conference outgoing dial* request is issued for a conference already started. As shown in the following figure, the target that receives the call is first prompted to join the conference. The *Meet-Me Participant Joined* event is issued only once the target has confirmed acceptance.

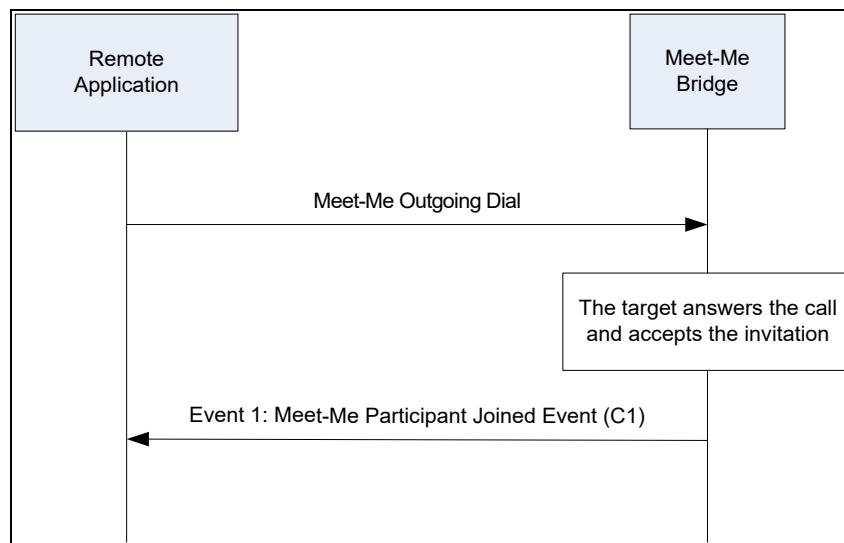


Figure 106 Meet-Me Conference Outgoing Dial Events for a Started Conference

In the second example, the outdial request is issued for a conference that has no participant and is therefore not started. As a result, once the target accepts to join the conference, a *Meet-Me Conference Started* event is first issued followed by a *Meet-Me Participant Joined* event.

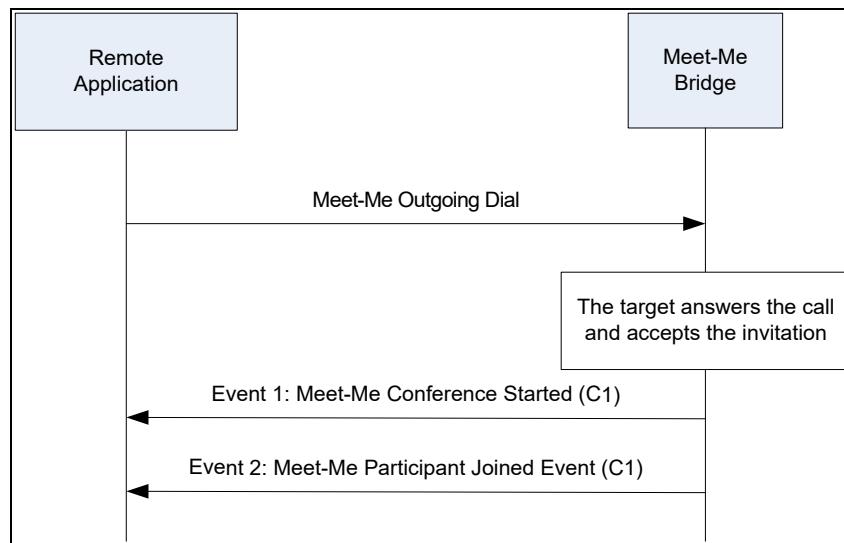


Figure 107 Meet-Me Conference Outgoing Dial Events for a New Conference

5.7.2.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110888	The target user is not a Meet-Me virtual subscriber.
400	110873	Invalid Conference ID.
400	110889	The conference has not started.
400	110893	The server hosting the conference is not available.
400	110872	The conference has expired.
400	110871	This conference is not active; Outdial failed.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

5.7.3 Mute Call

5.7.3.1 Description

This request mutes a particular participant on the call. When muted, no audio stream is transmitted from the muted participant associated with the call to the other participants.

5.7.3.2 Request

A remote application can issue this request by sending a MeetMeConferenceMuteCallRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>bridgeId</i>	SubscriberId	Yes	The subscriber ID of the Meet-Me conference bridge.
<i>conferenceId</i>	MeetMeConferenceId	Yes	The conference ID.
<i>callId</i>	CallId	Yes	The callId of the call to mute.

5.7.3.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.7.3.4 Associated Events

When the request is accepted, a *Meet-Me Participant Muted* event is generated if the call was not already muted. Otherwise, no event is generated.

5.7.3.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110888	The target user is not a Meet-Me virtual subscriber.
400	110873	Invalid Conference ID.

400	110889	The conference has not started.
400	110893	The server hosting the conference is not available.
400	110877	Participant does not exist.
400	110878	Participant on hold cannot be muted or unmuted.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.7.4 Unmute Call

5.7.4.1 Description

This request unmutes a particular participant on the call. When unmuted, the audio stream can again be transmitted from the conference participant associated with the call to the other participants.

5.7.4.2 Request

A remote application can issue this request by sending a MeetMeConferenceUnmuteCallRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>bridgeld</i>	SubscriberId	Yes	The subscriber ID of the Meet-Me conference bridge.
<i>conferenceId</i>	MeetMeConferenceld	Yes	The conference ID.
<i>callId</i>	CallId	Yes	The callId of the call to unmute.

5.7.4.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.7.4.4 Associated Events

When the request is accepted, a *Meet-Me Participant Unmuted* event is generated if the call was already muted. Otherwise, no event is generated.

5.7.4.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110888	The target user is not a Meet-Me virtual subscriber.
400	110873	Invalid Conference ID.
400	110889	The conference has not started.
400	110893	The server hosting the conference is not available.
400	110877	The participant does not exist.
400	110878	The participant on hold cannot be muted or unmuted.

400	110890	The number of unmuted participants has reached the maximum limit. You cannot unmute now.
-----	--------	---

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

For Call ID errors, see section [5.1.3.3 CallId Errors](#).

5.7.5 Hold Call

5.7.5.1 Description

This request puts a particular participant's call on hold.

5.7.5.2 Request

A remote application can issue this request by sending a MeetMeConferenceHoldCallRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>bridgeld</i>	SubscriberId	Yes	The subscriber ID of the Meet-Me conference bridge.
<i>conferenceId</i>	MeetMeConferenceld	Yes	The conference ID.
<i>callId</i>	CallId	Yes	The callId of the call to hold.

5.7.5.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.7.5.4 Associated Events

When the request is accepted, a *Meet-Me Participant Held* event is generated if the call was not already held. Otherwise, no event is generated.

5.7.5.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110888	The target user is not a Meet-Me virtual subscriber.
400	110873	Invalid Conference ID.
400	110889	The conference has not started.
400	110893	The server hosting the conference is not available.
400	110877	The participant does not exist.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

For Call ID errors, see section [5.1.3.3 CallId Errors](#).

5.7.6 Retrieve Call

5.7.6.1 Description

This request retrieves (unholds) a participant's call that was held.

5.7.6.2 Request

A remote application can issue this request by sending a MeetMeConferenceRetrieveCallRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>bridgeId</i>	SubscriberId	Yes	The subscriber ID of the Meet-Me conference bridge.
<i>conferenceId</i>	MeetMeConferenceId	Yes	The conference ID.
<i>callId</i>	CallId	Yes	The callId of the call to retrieve.

5.7.6.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.7.6.4 Associated Events

When the request is accepted, a *Meet-Me Participant Retrieved* event is generated if the call was already held. Otherwise, no event is generated.

5.7.6.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110888	The target user is not a Meet-Me virtual subscriber.
400	110873	Invalid Conference ID.
400	110889	The conference has not started.
400	110893	The server hosting the conference is not available.
400	110877	The participant does not exist.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.7.7 Release Call

5.7.7.1 Description

This request releases a participant's call from the conference.

The following diagram provides information on the state and topology changes involved.

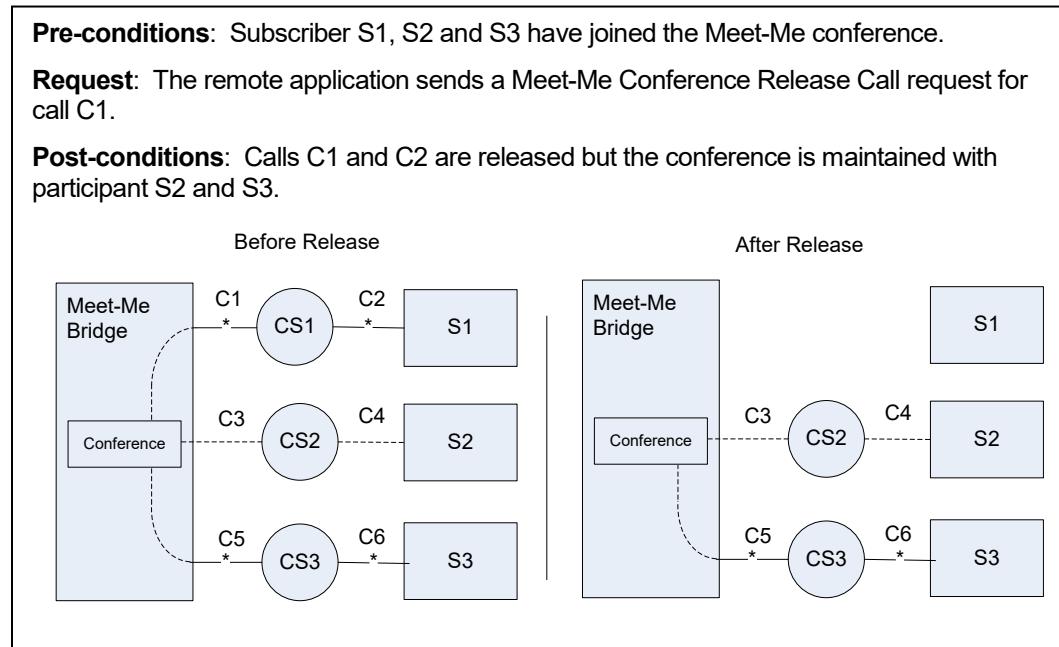


Figure 108 Meet-Me Release Call

5.7.7.2 Request

A remote application can issue this request by sending a `MeetMeConferenceReleaseCallRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>bridgeld</code>	SubscriberId	Yes	The subscriber ID of the Meet-Me conference bridge.
<code>conferenceld</code>	MeetMeConferenceld	Yes	The conference ID.
<code>callld</code>	CallId	Yes	The callId of the call to release.

5.7.7.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.7.7.4 Associated Events

When the request is accepted, a *Meet-Me Participant Left* event is generated.

5.7.7.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110888	The target user is not a Meet-Me virtual subscriber.
400	110873	Invalid Conference ID.
400	110889	The conference has not started.
400	110893	The server hosting the conference is not available.
400	110877	The participant does not exist.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.7.8 Lock

5.7.8.1 Description

This request locks a specified Meet-Me conference. Once the conference is locked, no participant can join the conference by dialing in to the conference bridge. The only way to add another participant is to use the Outgoing Dial request.

5.7.8.2 Request

A remote application can issue this request by sending a MeetMeConferenceLockRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>bridgeld</i>	SubscriberId	Yes	The subscriber ID of the Meet-Me conference bridge.
<i>conferenceld</i>	MeetMeConferenceld	Yes	The conference ID.

5.7.8.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.7.8.4 Associated Events

When the request is accepted, a *Meet-Me Conference Locked* event is generated.

5.7.8.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110888	The target user is not a Meet-Me virtual subscriber.
400	110873	Invalid Conference ID.
400	110889	The conference has not started.

statusCode	errorCode	Summary
400	110893	The server hosting the conference is not available.
400	110874	Another moderator level operation is in progress. Try later.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.7.9 Unlock

5.7.9.1 Description

This request unlocks a specified Meet-Me conference that was previously locked. Once successful, participants can join the conference by dialing in to the bridge.

5.7.9.2 Request

A remote application can issue this request by sending a MeetMeConferenceUnlockRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>bridgeId</i>	SubscriberId	Yes	The subscriber ID of the Meet-Me conference bridge.
<i>conferenceId</i>	MeetMeConferenceld	Yes	The conference ID.

5.7.9.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.7.9.4 Associated Events

When the request is accepted, a *Meet-Me Conference Unlocked* event is generated.

5.7.9.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110888	The target user is not a Meet-Me virtual subscriber.
400	110873	Invalid Conference ID.
400	110889	The conference has not started.
400	110893	The server hosting the conference is not available.
400	110874	Another moderator level operation is in progress. Try later.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.7.10 Start Recording

5.7.10.1 Description

This request can be used to start recording for a specified Meet-Me conference. An error response is returned if there is an attempt to start recording twice without stopping.

When the conference recording is stopped/started multiple times during a conference, a new recording is stored on the repository each time.

5.7.10.2 Request

A remote application can issue this request by sending a `MeetMeConferenceStartRecordingRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>bridgeld</code>	SubscriberId	Yes	The subscriber ID of the Meet-Me conference bridge.
<code>conferenceId</code>	MeetMeConferenceId	Yes	The conference ID.

5.7.10.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.7.10.4 Associated Events

When the request is accepted, a *Meet-Me Conference Recording Started* event is generated.

5.7.10.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110888	The target user is not a Meet-Me virtual subscriber.
400	110873	Invalid Conference ID.
400	110889	The conference has not started.
400	110893	The server hosting the conference is not available.
400	110882	Recording is not enabled for this conference.
400	110891	Recording cannot be started until the conference has started.
400	110874	Another moderator level operation is in progress. Try later.
400	110870	The Profile Server address is either inactive or invalid.
400	110885	This conference recording is paused; it can either resume or stop.
400	110883	This conference is already being recorded.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.7.11 Pause Recording

5.7.11.1 Description

This request can be used to pause a recording for the Meet-Me Conference that is being recorded.

5.7.11.2 Request

A remote application can issue this request by sending a `MeetMeConferencePauseRecordingRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>bridgeld</code>	SubscriberId	Yes	The subscriber ID of the Meet-Me conference bridge.
<code>conferenceld</code>	MeetMeConferenceld	Yes	The conference ID.

5.7.11.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.7.11.4 Associated Events

When the request is accepted, a *Meet-Me Conference Recording Paused* event is generated.

5.7.11.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110888	The target user is not a Meet-Me virtual subscriber.
400	110873	Invalid Conference ID.
400	110889	The conference has not started.
400	110893	The server hosting the conference is not available.
400	110882	Recording is not enabled for this conference.
400	110874	Another moderator level operation is in progress. Try later.
400	110886	This conference is not being recorded.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		



5.7.12 Resume Recording

5.7.12.1 Description

This request can be used to resume recording for the Meet-Me Conference that was paused earlier.

5.7.12.2 Request

A remote application can issue this request by sending a `MeetMeConferenceResumeRecordingRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>bridgeld</code>	<code>SubscriberId</code>	Yes	The subscriber ID of the Meet-Me conference bridge.
<code>conferenceId</code>	<code>MeetMeConferenceld</code>	Yes	The conference ID.

5.7.12.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.7.12.4 Associated Events

When the request is accepted, a *Meet-Me Conference Recording Resumed* event is generated.

5.7.12.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110888	The target user is not a Meet-Me virtual subscriber.
400	110873	Invalid Conference ID.
400	110889	The conference has not started.
400	110893	The server hosting the conference is not available.
400	110882	Recording is not enabled for this conference.
400	110874	Another moderator level operation is in progress. Try later.
400	110886	This conference is not being recorded.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.7.13 Stop Recording

5.7.13.1 Description

This request can be used to stop recording for the Meet-Me Conference that was started earlier. If the recording was not started, this command returns an error. When the recording is stopped, the recording is uploaded to the repository.

5.7.13.2 Request

A remote application can issue this request by sending a `MeetMeConferenceStopRecordingRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>bridgeld</code>	SubscriberId	Yes	The subscriber ID of the Meet-Me conference bridge.
<code>conferenceId</code>	MeetMeConferenceld	Yes	The conference ID.

5.7.13.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.7.13.4 Associated Events

When the request is accepted, a *Meet-Me Conference Recording Stopped* event is generated.

5.7.13.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110888	The target user is not a Meet-Me virtual subscriber.
400	110873	Invalid Conference ID.
400	110889	The conference has not started.
400	110893	The server hosting the conference is not available.
400	110882	Recording is not enabled for this conference.
400	110874	Another moderator level operation is in progress. Try later.
400	110886	This conference is not being recorded.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.7.14 Set Lecture Mode

5.7.14.1 Description

This request can be used to set the lecture mode for a specified Meet-Me conference. Once the Lecture Mode is set with a particular moderator, only the moderator can talk and all the other participants in the conference are muted.

5.7.14.2 Request

A remote application can issue this request by sending a `MeetMeConferenceSetLectureModeRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>bridgeld</code>	SubscriberId	Yes	The subscriber ID of the Meet-Me conference bridge.
<code>conferenceId</code>	MeetMeConferenceId	Yes	The conference ID.

5.7.14.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.7.14.4 Associated Events

When the request is accepted, a *Meet-Me Conference Lecture Mode Started* event is generated if the conference is not already in lecture mode. Otherwise, no event is generated.

5.7.14.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110888	The target user is not a Meet-Me virtual subscriber.
400	110873	Invalid Conference ID.
400	110889	The conference has not started.
400	110893	The server hosting the conference is not available.
400	110874	Another moderator level operation is in progress. Try later.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.7.15 Unset Lecture Mode

5.7.15.1 Description

This request can be used to set the lecture mode for a specified Meet-Me conference.

5.7.15.2 Request

A remote application can issue this request by sending a `MeetMeConferenceUnsetLectureModeRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>bridgeld</code>	SubscriberId	Yes	The subscriber ID of the Meet-Me conference bridge.
<code>conferenceId</code>	MeetMeConferenceId	Yes	The conference ID.

5.7.15.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.7.15.4 Associated Events

When the request is accepted, a *Meet-Me Conference Lecture Mode Stopped* event is generated if the conference is in lecture mode. Otherwise, no event is generated.

5.7.15.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110888	The target user is not a Meet-Me virtual subscriber.
400	110873	Invalid Conference ID.
400	110889	The conference has not started.
400	110893	The server hosting the conference is not available.
400	110874	Another moderator level operation is in progress. Try later.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).



5.7.16 Play Security Classification

5.7.16.1 Description

This request can be used to play a message that announces the current security classification level calculated for a specified Meet-Me conference.

5.7.16.2 Request

A remote application can issue this request by sending a MeetMeConference PlaySecurityClassificationRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>bridgeld</i>	SubscriberId	Yes	The subscriber ID of the Meet-Me conference bridge.
<i>conferenceId</i>	MeetMeConferenceId	Yes	The conference ID.

5.7.16.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.7.16.4 Associated Events

No event is generated.

5.7.16.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110873	Invalid Conference ID.
400	110888	Target User is not a Meet-Me virtual subscriber.
400	110889	Conference has not started.
400	110893	The server hosting the conference is not available.
400	111043	The conference does not have a security classification.
400	110044	The security classification announcement cannot be played when there is only one participant.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

5.7.17 Set Conference Client Session Info

5.7.17.1 Description

This request is used by a remote application to explicitly associate the client session info with the existing Meet-Me conference. Upon receiving this command, Cisco BroadWorks stores the client session info against all call sessions connected to the Meet-Me conference focus prior to the reception of this command. Cisco BroadWorks also updates the call session of any new call connected to the Meet-Me conference focus after the reception of this command. Cisco BroadWorks triggers notifications to the remote end of those call sessions. The notifications are sent over the SIP and/or CTI interfaces.

There is no state or topology change that occurs when the request is invoked.

5.7.17.2 Request

A remote application can issue this request by sending a `MeetMeConferenceSetConferenceClientSessionInfoRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>bridgeId</code>	SubscriberId	Yes	The subscriber ID of the Meet-Me conference bridge.
<code>conferenceId</code>	MeetMeConferenceId	Yes	The conference ID.
<code>Info</code>	String	Yes	This is opaque information provided by the client for the Application Server to correlate this information with the call session. The Application Server stores this information against the call session and triggers notifications for both ends of the call session over the SIP or CTI interface.

5.7.17.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.7.17.4 Associated Events

The `MeetMeConferenceClientSessionInfoUpdatedEvent` is sent when the client session info is successfully set.

5.7.17.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110873	Invalid Conference ID.
400	110888	Target User is not a Meet-Me virtual subscriber.
400	110889	Conference has not started.
400	110893	The server hosting the conference is not available.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.8 Collaborate Room Requests (**AS only**)

5.8.1 Get Collaborate Room

5.8.1.1 Description

This request is used to get details about an active Collaborate Room. The information provided contains details regarding the room itself as well as the list of participants who joined the room.

5.8.1.2 Request

A remote application can issue this request by sending a GetCollaborateRoomRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the room owner.
<i>roomId</i>	CollaborateRoomId	Yes	The room ID.

5.8.1.3 Response

The response contains the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>CollaborateRoom</i>	CollaborateRoom	No	The detail of the Collaborate room. Only present if the statusCode was set to "200".

5.8.1.4 Associated Events

No events are associated with this request.

5.8.1.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111200	The user does not have the Collaborate – Audio service assigned.
400	111201	The room <roomid> does not exist.
For <i>System</i> errors, see section 5.1.3.5 System Errors .		
For <i>Target User</i> errors, see section 5.1.3.1 Target User Errors .		

5.8.2 Outgoing Dial

5.8.2.1 Description

This request is used to originate a call session from the Collaborate Room to the specified destination. If the request is accepted, then Cisco BroadWorks initiates a call to the target destination.

The target is prompted to join the room once the call is answered. The call is considered to have joined the room only once the target has confirmed acceptance. The target does not appear to have a participant until then.

The following diagram provides information on the state and topology changes involved.

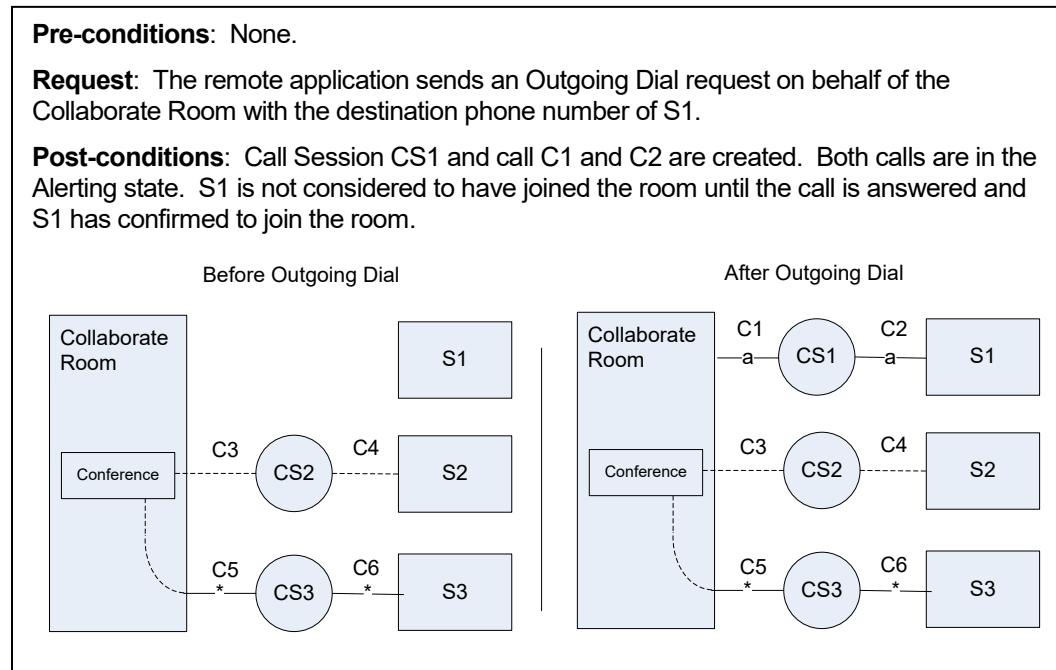


Figure 109 Collaborate Outgoing Dial

5.8.2.2 Request

A remote application can issue this request by sending a `CollaborateRoomOutgoingDialRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID of the room owner.
<code>roomId</code>	CollaborateRoomId	Yes	The room ID.
<code>address</code>	Address	Yes	The address of the party to add to the room.

5.8.2.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.8.2.4 Associated Events

The following time sequence diagram illustrates the event produced when a *Collaborate Room outgoing dial* request is issued for a room already started. As shown in the following figure, the target that receives the call is first prompted to join the room. The Collaborate Room Participant Joined event is issued only once the target has confirmed acceptance.

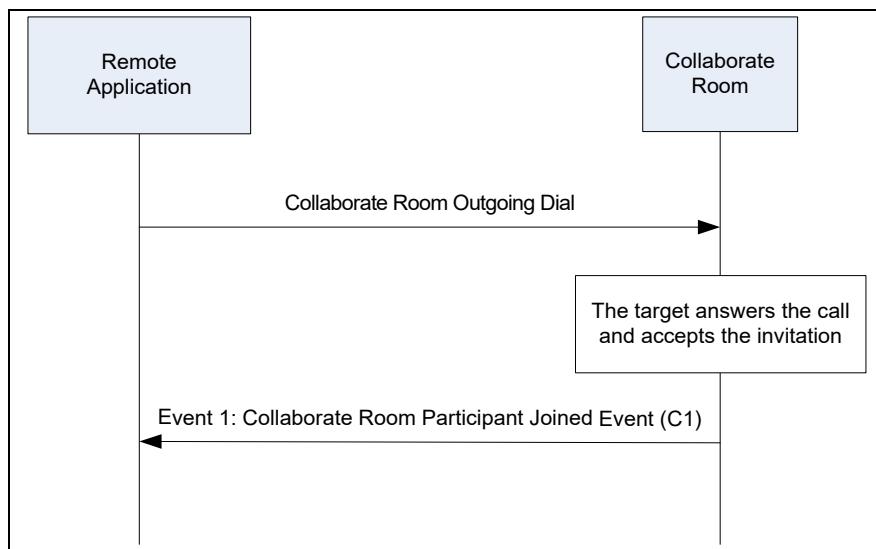


Figure 110 Collaborate Room Outgoing Dial Events for a Started Conference

In the second example, the outdial request is issued for a room that has no participant and is therefore not started. As a result, once the target accepts to join the room, a *Collaborate Room Started* event is first issued followed by a *Collaborate Room Participant Joined* event.

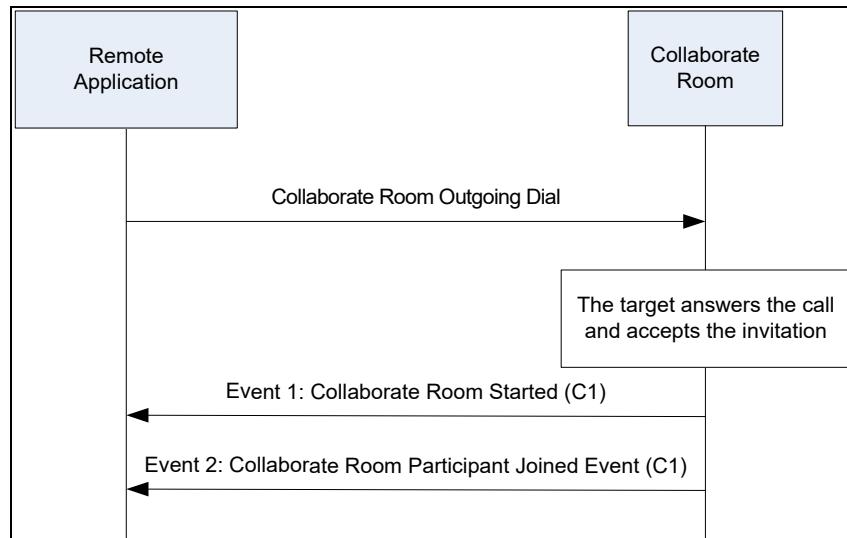


Figure 111 Collaborate Room Outgoing Dial Events for a New Conference

5.8.2.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111200	The user does not have the Collaborate – Audio service assigned.
400	111201	The room <roomid> does not exist.
400	111203	The room is locked. Ask the owner to unlock the room.
400	111206	Collaborate Room Not Started.
400	111207	Collaborate Room Hosting Server not available.
400	111208	Collaborate Room has Expired.
400	111209	Collaborate Room Participant declined outdial request.
400	111210	Invalid Outdial Address.
400	111212	Room has maximum number of participants. Unable to add participant.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

5.8.3 Mute Call

5.8.3.1 Description

This request mutes a particular participant on the call. When muted, no audio stream is transmitted from the muted participant associated with the call to the other participants.

5.8.3.2 Request

A remote application can issue this request by sending a CollaborateRoomMuteCallRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the room owner.
<i>roomId</i>	CollaborateRoomId	Yes	The room ID.
<i>callId</i>	CallId	Yes	The callId of the call to mute.

5.8.3.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.8.3.4 Associated Events

When the request is accepted, a *Collaborate Room Participant Muted* event is generated if the call was not already muted. Otherwise, no event is generated.

5.8.3.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111200	The user does not have the Collaborate – Audio service assigned.
400	111201	The room <roomid> does not exist.
400	111211	The participant does not exist.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.8.4 Unmute Call

5.8.4.1 Description

This request unmutes a particular participant on the call. When unmuted, the audio stream can again be transmitted from the room participant associated with the call to the other participants.

5.8.4.2 Request

A remote application can issue this request by sending a CollaborateUnmuteCallRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the room owner.
<i>roomId</i>	CollaborateRoomId	Yes	The room ID.
<i>callId</i>	CallId	Yes	The callId of the call to unmute.

5.8.4.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.8.4.4 Associated Events

When the request is accepted, a *Collaborate Room Participant Unmuted* event is generated if the call was already muted. Otherwise, no event is generated.

5.8.4.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111200	The user does not have the Collaborate – Audio service assigned.
400	111201	The room <roomid> does not exist.
400	111211	The participant does not exist.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.8.5 Release Call

5.8.5.1 Description

This request releases a participant's call from the room.

The following diagram provides information on the state and topology changes involved.

Pre-conditions: Subscriber S1, S2, and S3 have joined the Collaborate Room.

Request: The remote application sends a Collaborate Room Release Call request for call C1.

Post-conditions: Calls C1 and C2 are released but the room is maintained with participant S2 and S3.

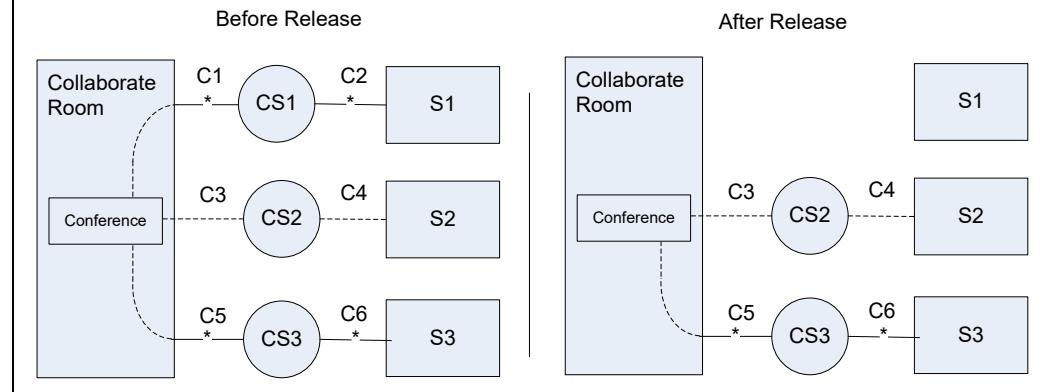


Figure 112 Collaborate Room Release Call

5.8.5.2 Request

A remote application can issue this request by sending a `CollaborateRoomReleaseCallRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	<code>SubscriberId</code>	Yes	The subscriber ID of the room owner.
<code>roomId</code>	<code>CollaborateRoomId</code>	Yes	The room ID.
<code>callId</code>	<code>CallId</code>	Yes	The callId of the call to release.

5.8.5.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.8.5.4 Associated Events

When the request is accepted, a *Collaborate Room Participant Left* event is generated.

5.8.5.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111200	The user does not have the Collaborate – Audio service assigned.
400	111201	The room <roomid> does not exist.
400	111206	Collaborate Room Not Started.
400	111207	Collaborate Room Hosting Server not available.
400	111211	The participant does not exist.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Call ID errors, see section 5.1.3.3 CallId Errors .		

5.8.6 Lock

5.8.6.1 Description

This request locks a specified active Collaborate Room. Once the room is locked, no participant can join the conference by dialing in to the room or by using the Outgoing Dial request.

5.8.6.2 Request

A remote application can issue this request by sending a CollaborateRoomLockRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
subscriberId	SubscriberId	Yes	The subscriber ID of the room owner.
roomId	CollaborateRoomId	Yes	The room ID.

5.8.6.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.8.6.4 Associated Events

When the request is accepted, a *Collaborate Room Locked* event is generated.

5.8.6.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111200	The user does not have the Collaborate – Audio service assigned.

statusCode	errorCode	Summary
400	111201	The room <roomid> does not exist.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.8.7 Unlock

5.8.7.1 Description

This request unlocks a specified active Collaborate that was previously locked. Once successful, participants can join the room by dialing in to the bridge.

5.8.7.2 Request

A remote application can issue this request by sending a CollaborateRoomUnlockRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
subscriberId	SubscriberId	Yes	The subscriber ID of the room owner.
roomId	CollaborateRoomId	Yes	The room ID.

5.8.7.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.8.7.4 Associated Events

When the request is accepted, a *Collaborate Room Unlocked* event is generated.

5.8.7.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111200	The user does not have the Collaborate – Audio service assigned.
400	111201	The room <roomid> does not exist.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.8.8 Set Conference Client Session Info

5.8.8.1 Description

This request is used by a remote application to explicitly associate the client session info with the specified active Collaborate Room. Upon receiving this command, Cisco BroadWorks stores the client session info against all call sessions connected to the Collaborate Room focus prior to the reception of this command. Cisco BroadWorks also updates the call session of any new call connected to the Collaborate Room focus after the reception of this command. Cisco BroadWorks triggers notifications to the remote end of those call sessions. The notifications are sent over the SIP and/or CTI interfaces.

There is no state or topology change that occurs when the request is invoked.

5.8.8.2 Request

A remote application can issue this request by sending a `CollaborateRoomSetClientSessionInfoRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID of the room owner.
<code>roomId</code>	CollaborateRoomId	Yes	The room ID.
<code>Info</code>	String	Yes	This is opaque information provided by the client for the Application Server to correlate this information with the call session. The Application Server stores this information against the call session and triggers notifications for both ends of the call session over the SIP or CTI interface.

5.8.8.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.8.8.4 Associated Events

The `CollaborateRoomClientSessionInfoUpdatedEvent` is sent when the client session info is successfully set.

5.8.8.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	111200	The user does not have the Collaborate – Audio service assigned.
400	111201	The room <roomid> does not exist.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.9 Service Management Requests

This section describes the service management requests. Note that if a response returned by a service management request contains a user ID, this user ID may not have a domain part. In such a case, the domain must be considered equal to the system default domain.

5.9.1 Modify Call Forwarding Always

5.9.1.1 Description

This request is used to set or modify the Call Forwarding Always (CFA) service configuration of a subscriber. CFA enables a subscriber to redirect all incoming calls unconditionally (that is, busy, idle, alerting, and so on) to another phone number. If activated, a subscriber must specify the forwarding number. Optionally, a ring splash (500 ms ring burst) can be configured to the subscriber's device each time a call is forwarded.

This request does not involve any call state or topology changes.

5.9.1.2 Request

A remote application can issue this request by sending a `ModifyCallForwardingAlways` Request. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID of the user for which the Call Forwarding Always service is modified.
<code>callForwardingAlways</code>	CallForwardingAlways	Yes	The associated Call Forwarding Always configuration.

5.9.1.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.9.1.4 Associated Events

When a Modify Call Forwarding Always request is accepted, a *Call Forwarding Always* event is issued against the subscriber.

5.9.1.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
401	4008	User not found: <userId>.
401	4003	Unauthorized request: <request>.
400	4410	Service is not assigned to this subscriber: <userId>.
400	5700	This service requires a phone number to turn the service on.
400	4981	Invalid phone number. An E.164 number used for outgoing calls must contain only characters +, -, and 0-9. Only the leading character can be a "+".

statusCode	errorCode	Summary
400	4807	Invalid phone number. A phone number used for outgoing calls must contain only -, 0-9, or an E.164 formatted number. Additional spaces and dashes are allowed but are not included in the number.
400	4980	Invalid phone number. An E.164 number used for outgoing calls must be 3 to 22 characters in length. Dashes are not included in this length. There must be at least one digit after the country code.
400	4453	Invalid character(s) found in feature access code. A feature access code can contain digits and the characters: *, #, A, B, C, D: <AccessCode>.
400	4414	Invalid number. The number entered is an emergency number: <userId>.
400	4420	Invalid number. The number entered is a repair number: <userId>
400	4429	Invalid number. The number entered is a chargeable directory assistance number: <userId>.
400	4024	Invalid country code specified: <userId>.
400	4806	Invalid phone number. A phone number used for outgoing calls must be 2 to 30 digits. Spaces and dashes are not included in this length.

For System errors, see section [5.1.3.5 System Errors](#).

5.9.2 Get Call Forwarding Always

5.9.2.1 Description

This request is used to get the Call Forwarding Always service configuration of a subscriber.

This request does not involve any call state or topology changes.

5.9.2.2 Request

A remote application can issue this request by sending a GetCallForwardingAlwaysRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the user for which the Call Forwarding Always configuration is queried.

5.9.2.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>callForwardingAlways</i>	CallForwardingAlways	No	The associated Call Forwarding Always attributes. Only present when a successful response is returned (that is, statusCode is set to "200").

5.9.2.4 Associated Events

No event is associated with this request.

5.9.2.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
401	4008	User not found: <userId>.
401	4003	Unauthorized request: <request>.
400	4410	Service is not assigned to this subscriber: <userId>.
For System errors, see section 5.1.3.5 System Errors .		

5.9.3 Modify Call Forwarding No Answer

5.9.3.1 Description

This request is used to set or modify the Call Forwarding No Answer service configuration of a subscriber. Call Forwarding No Answer (CFNA) enables a subscriber to redirect all unanswered incoming calls to another phone number. A call is considered unanswered if it is in the *Alerting* state for a configured duration (specified in number of localized ringback duration). The maximum number of rings is 20 (inclusive).

This request does not involve any call state or topology changes.

5.9.3.2 Request

A remote application can issue this request by sending a `ModifyCallForwardingNoAnswer` Request. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID of the user for which the Call Forwarding No Answer service is modified.
<code>callForwardingAlways</code>	CallForwardingAlways	Yes	The associated Call Forwarding No Answer configuration.

5.9.3.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.9.3.4 Associated Events

When a `Modify Call Forwarding Always` request is accepted, a `Call Forward No Answer` event is issued against the subscriber.

5.9.3.5 Error Cases

The same error listed for the `Modify Call Forwarding Always` request applies to the `Modify Call Forwarding No Answer` request.

5.9.4 Get Call Forwarding No Answer

5.9.4.1 Description

This request is used to get the Call Forwarding No Answer (CFNA) service configuration of a subscriber. This request does not involve any call state or topology changes.

5.9.4.2 Request

A remote application can issue this request by sending a GetCallForwardingNoAnswer Request. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the user for which the Call Forwarding No Answer configuration is queried.

5.9.4.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>callForwardingNoAnswer</i>	CallForwardingNoAnswer	No	The associated Call Forwarding No Answer attributes. Only present when a successful response is returned (that is, the statusCode is set to "200").

5.9.4.4 Associated Events

No event is associated with this request.

5.9.4.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
401	4008	User not found: <userId>.
401	4003	Unauthorized request: <request>.
400	4410	Service is not assigned to this subscriber: <userId>.
For System errors, see section 5.1.3.5 System Errors .		

5.9.5 Modify Call Forwarding Busy

5.9.5.1 Description

This request is used to set or modify the Call Forwarding Busy service configuration of a subscriber. Call Forwarding Busy (CFB) enables a subscriber to redirect incoming calls to another phone number when the subscriber is busy. A subscriber is considered busy when there are too many active calls or a service makes the subscriber appear busy to the caller (for example, services such as Do Not Disturb or Selective Call Rejection).

This request does not involve any call state or topology changes.

5.9.5.2 Request

A remote application can issue this request by sending a `ModifyCallForwardingBusyRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID of the user for which the Call Forwarding Busy service is modified.
<code>callForwardingBusy</code>	CallForwardingBusy	Yes	The associated Call Forwarding Busy configuration.

5.9.5.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.9.5.4 Associated Events

When a Modify Call Forwarding Always request is accepted, a Call Forward Always Busy event is issued against the subscriber.

5.9.5.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
401	4008	User not found: <userId>.
401	4003	Unauthorized request: <request>.
400	4410	Service is not assigned to this subscriber: <userId>.
400	5700	This service requires a phone number to turn the service on.
400	4981	Invalid phone number. An E.164 number used for outgoing calls must contain only characters +, -, and 0-9. Only the leading character can be a "+".
400	4807	Invalid phone number. A phone number used for outgoing calls must contain only -, 0-9, or an E.164 formatted number. Additional spaces and dashes are allowed but cannot be included in the number.
400	4980	Invalid phone number. An E.164 number used for outgoing calls must be 3 to 22 characters in length. Dashes are not included in this length. There must be at least one digit after the country code.
400	4453	Invalid character(s) found in feature access code. A feature access code can contain digits and characters *, #, A, B, C, D: <AccessCode>.

statusCode	errorCode	Summary
400	4414	Invalid number. The number entered is an emergency number: <number>.
400	4420	Invalid number. The number entered is a repair number: <number>.
400	4429	Invalid number. The number entered is a chargeable directory assistance number: <number>.
400	4024	Invalid country code specified: <number>.
400	4806	Invalid phone number. A phone number used for outgoing calls must be 2 to 30 digits. Spaces and dashes are not included in this length.

For System errors, see section [5.1.3.5 System Errors](#).

5.9.6 Get Call Forwarding Busy

5.9.6.1 Description

This request is used to get the Call Forwarding Busy (CFB) service configuration of a subscriber.

This request does not involve any call state or topology changes.

5.9.6.2 Request

A remote application can issue this request by sending a GetCallForwardingBusyRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the user for which the Call Forwarding Busy configuration is queried.

5.9.6.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>callForwardingBusy</i>	CallForwardingBusy	No	The associated Call Forwarding Busy attributes. Only present when a successful response is returned (that is, the statusCode is set to "200").

5.9.6.4 Associated Events

No event is associated with this request.

5.9.6.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
401	4008	User not found: <userId>.
401	4003	Unauthorized request: <request>.
400	4410	Service is not assigned to this subscriber: <userId>.

For System errors, see section [5.1.3.5 System Errors](#).

5.9.7 Modify Hoteling Guest

5.9.7.1 Description

This request is used to set or modify the Hoteling Guest configuration. The subscriber must have the Hoteling Guest service. This request is used to establish and delete the association between the Hoteling Guest and its Host.

The subscriber Hoteling Guest configuration specifies:

- If the subscriber can be associated with a host subscriber account.
- If the subscriber is currently associated with a host.
- If there is a time limit to a subscriber association.

The associate between the guest and the subscriber is done through this request. For more information on Hoteling, see section [4.8.1.17 Hoteling](#).

A remote application can clear the association between the host and the guest by setting the hostUserId of the hotelingGuest parameters to "nil".

This request does not involve any call state or topology changes.

5.9.7.2 Request

A remote application can issue this request by sending a ModifyHotelGuestRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the subscriber for which the guest service is modified.
<i>hotelngGuest</i>	HotelngGuest	Yes	The associated guest configuration.

5.9.7.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.9.7.4 Associated Events

When a Modify Hoteling Guest request is accepted, a *Hotelng Guest* event is issued against the subscriber.

5.9.7.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
401	4008	User not found: <userId>.
401	4003	Unauthorized request: <request>.
400	4410	Service is not assigned to this subscriber: <userId>.
400	5008	Host access level cannot be set to enterprise. The user does not belong to an enterprise.

For System errors, see section [5.1.3.5 System Errors](#).

5.9.8 Get Hoteling Host List (AS only)

5.9.8.1 Description

This request is used to get for a subscriber the list of available Hoteling Host endpoints to use with the Hoteling Guest service.

This request does not involve any call state or topology changes.

5.9.8.2 Request

A remote application can issue this request by sending a GetHotelngHostListRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the user for which the Hoteling Guest configuration is queried.

5.9.8.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>availableHotelngHosts</i>	AvailableHotelngHosts	No	The list of hosts available. Only present when a successful response is returned (that is, the statusCode is set to "200").

5.9.8.4 Associated Events

No event is associated with this request.

5.9.8.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
401	4008	User not found: <userId>.
401	4003	Unauthorized request: <request>.
400	4410	Service is not assigned to this subscriber: <userId>.
For System errors, see section 5.1.3.5 System Errors .		

5.9.9 Get Hoteling Guest (AS only)

5.9.9.1 Description

This request is used to get the Hoteling Guest service configuration of a subscriber.

This request does not involve any call state or topology changes.

5.9.9.2 Request

A remote application can issue this request by sending a GetHotelngGuestRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the user for which the Hoteling Guest configuration is queried.

5.9.9.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>hotelngGuest</i>	HotelingGuest	No	The associated guest configuration. Only present when a successful response is returned (that is, the statusCode is set to "200").

5.9.9.4 Associated Events

No event is associated with this request.

5.9.9.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
401	4008	User not found: <userId>.
401	4003	Unauthorized request: <request>.
400	4410	Service is not assigned to this subscriber: <userId>.

statusCode	errorCode	Summary
------------	-----------	---------

For System errors, see section [5.1.3.5 System Errors](#).

5.9.10 Modify Do Not Disturb

5.9.10.1 Description

This request is used to set or modify the Do Not Disturb (DND) service. This service allows a subscriber to set their status as unavailable.

This request does not involve any call state or topology changes.

5.9.10.2 Request

A remote application can issue this request by sending a `ModifyDoNotDisturbRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID of the subscriber for which Do Not Disturb service is modified.
<code>doNotDisturb</code>	DoNotDisturb	Yes	The associated Do Not Disturb configuration.

5.9.10.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.9.10.4 Associated Events

When a Modify Do Not Disturb request is accepted, a Do Not Disturb event is issued against the subscriber.

5.9.10.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
401	4008	User not found: <userId>.
401	4003	Unauthorized request: <request>.
400	4410	Service is not assigned to this subscriber: <userId>.

For System errors, see section [5.1.3.5 System Errors](#).

5.9.11 Get Do Not Disturb

5.9.11.1 Description

This request is used to get the Do Not Disturb service configuration of a subscriber.

This request does not involve any call state or topology changes.

5.9.11.2 Request

A remote application can issue this request by sending a GetDoNotDisturbRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the subscriber for which Do Not Disturb configuration is queried.

5.9.11.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>doNotDisturb</i>	DoNotDisturb	No	The associated Do Not Disturb configuration. Only present when a successful response is returned (that is, the statusCode is set to "200").

5.9.11.4 Associated Events

No event is associated with this request.

5.9.11.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
401	4008	User not found: <userId>.
401	4003	Unauthorized request: <request>.
400	4410	Service is not assigned to this subscriber: <userId>.
For System errors, see section 5.1.3.5 System Errors .		

5.9.12 Clear Voice Mail Message Summary

5.9.12.1 Description

This request is used to clear the voice mail message summary of the Third-Party Voice Mail Support or Voice Messaging service of a subscriber. When the voice mail message summary is cleared, the number of messages is set to "0" for all message types. The subscriber must have one of these services to use this request.

This request does not involve any call state or topology changes.

5.9.12.2 Request

A remote application can issue this request by sending a ClearVoiceMailMessageSummary Request. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID associated with the voice mail message summary to be cleared.

5.9.12.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.9.12.4 Associated Events

When a Clear Voice Mail Message Summary request is successfully accepted, a *Voice Mail Message Summary* event is sent to report the change to the voice mail.

5.9.12.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110800	User is not assigned the Voice Messaging User service, Third-Party Voice Mail Support service or the service is not enabled.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.9.13 Get Voice Mail Message Summary

5.9.13.1 Description

This request is used to get the voice mail message summary of the Third-Party Voice Mail.

The Support/Voice Messaging service is for the specified subscriber. If the subscriber does not have either service active, then the response always has an empty message summary.

This request does not involve any call state or topology changes.

5.9.13.2 Request

A remote application can issue this request by sending a GetVoiceMailMessageSummaryRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the subscriber for which the message summary is queried.

5.9.13.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
voiceMailMessageSummary	voiceMailMessageSummary	No	The associated voice mail message summary. Only present when a successful response is returned (that is, the statusCode is set to "200").

5.9.13.4 Associated Events

No event is associated with this request.

5.9.13.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
		For System errors, see section 5.1.3.5 System Errors .
		For Target User errors, see section 5.1.3.1 Target User Errors .

5.10 Event Channel Requests

5.10.1 Add Channel

5.10.1.1 Description

This request is used to add an event channel on behalf of the authenticated subscriber in Cisco BroadWorks. The channel is associated to the connection over which it is created and to an event channel set identifier provided by the remote application.

Associated with each channel is a priority, weight, applicationId, and expires value. The priority defines the order of the event channel within a set and weight defines the load balancing factor amongst channels with the same priority in a set. The applicationId enforces the limit of a single channel set per application. The expires value defines the life of the event channel. For more information on these parameters, see section [3.4 Events](#).

The expiration value must be within the range configured in the system. If the provided value exceeds the maximum allowed value, then the expiration value is set to the maximum allowed value. If the provided expiration value is less than 3600 (one hour), then the value is set to "3600".

If the request is accepted, the Cisco BroadWorks assigns to the event channel a channelId and returns it in the response.

An application can add more than one event channel over the same connection.

5.10.1.2 Request

A remote application can issue this request by sending an AddChannelRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>channel</i>	Channel	Yes	The details of the event channel to add.

5.10.1.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>channel</i>	Channel	No	Contains the channel created. Only present when a successful response is returned (that is, the statusCode is set to "200").

5.10.1.4 Associated Events

There is no event associated with this request.

5.10.1.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
For System errors, see section 5.1.3.5 System Errors .		

5.10.2 Update Channel

5.10.2.1 Description

This request is used to modify an event channel. It is used to refresh a channel prior to its expiration.

The expiration value must be within the range configured in the system. If the provided value exceeds the maximum allowed value, then the expiration value is set to the maximum allowed value. If the provided expiration value is less than 3600 (one hour), then the value is set to "3600".

5.10.2.2 Request

A remote application can issue this request by sending an UpdateChannelRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>channel</i>	Channel	Yes	The details of the event channel to update. Only the channelId and expires parameters in the channel are taken into consideration, other parameters are ignored.

5.10.2.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>channel</i>	Channel	No	Contains the detail of the updated channel. Only present when a successful response is returned (that is, the statusCode is set to "200").

5.10.2.4 Associated Events

There is no event associated with this request.

5.10.2.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
403	110841	Channel <channelId> not found.
For System errors, see section 5.1.3.5 System Errors .		

5.10.3 Delete Channel

5.10.3.1 Description

This request is used to delete an event channel.

5.10.3.2 Request

A remote application can issue this request by sending a DeleteChannelRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>channelId</i>	ChannelId	Yes	The channel ID of the channel to be deleted.

5.10.3.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.10.3.4 Associated Events

No event is associated with this request.

5.10.3.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
403	110841	Channel <channelId> not found.
For System errors, see section 5.1.3.5 System Errors .		

5.10.4 Get Channel Set

5.10.4.1 Description

This request is used to request the set of event channels for the authenticated subscriber and an event channel set identifier. If the request is accepted and one or multiple channels are found, then the returned response contains the list of channels.

5.10.4.2 Request

A remote application can issue this request by sending a GetChannelSetRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>channelSetId</i>	ChannelSetId	Yes	The channelset ID used to query the event channel set, other parameters are ignored.

5.10.4.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>channelSet</i>	ChannelSet	No	The set of event channels found. Only present when a successful response is returned (that is, the statusCode is set to "200") and at least one channel was found with the <i>channelSetId</i> specified in the request.

5.10.4.4 Associated Events

There is no event associated with this request.

5.10.4.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
403	110842	Channel set <channelSetId> not found.
For System errors, see section 5.1.3.5 System Errors .		

5.11 Event Subscription Requests

5.11.1 Add User Subscription

5.11.1.1 Description

This request is used to add a subscription against the specified user on behalf of the authenticated subscriber in Cisco BroadWorks. This subscription is for an event package, and triggers notifications back to the issuing remote client application when those events are triggered. The subscription is associated with an event channel set using the set identifier. An event channel must have been created for this event channel set; otherwise, the request is rejected.

If an “Add User Subscription” request is received to create a subscription and there is already an existing subscription with the same *subscriberId*, *targetId*, *event*, and *applicationId*, the second subscription is considered a duplicate of the first. Cisco BroadWorks then issues a *Subscription Terminated* event on the first subscription, and accepts the second subscription in place of the first one.

The expiration value must be within the range configured in the system. If the provided value exceeds the maximum allowed value, then the expiration value is set to the maximum allowed value. If the provided expiration value is less than 3600 (one hour), then the value is set to “3600”.

5.11.1.2 Request

A remote application can issue this request by sending an *AddUserSubscriptionRequest*. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the subscription target.
<i>subscription</i>	Subscription	Yes	The detail of the subscription to add.

Only the basic elements relevant for a new subscription should be provided by the *subscription* parameter. The following elements are ignored and should not be provided with this request:

- *subscriptionId*
- *subscriberId*
- *targetIdType*
- *targetGrpParentId*
- *targetId*
- *targetSubId*

5.11.1.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscription</i>	Subscription	No	Contains the detail of the created subscription. Only present when a successful response is returned (that is, the statusCode is set to "200").

5.11.1.4 Associated Events

The *Subscription Terminated* event can be received following a subscription creation.

An initial notification is sent for the subscription (if the event package does not support targetSubId subscription).

5.11.1.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
403	110842	Channel set <channelSetId> not found.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Subscription errors, see section 5.1.3.6 Subscription Errors .		

5.11.2 Add SubUser Subscription

5.11.2.1 Description

This request is used to add a subscription at the subId level. This subscription is for an event package that supports target subId (like Meet-Me conference or Collaborate Room), and triggers notifications back to the issuing remote client application when those events are triggered. The subscription is associated with an event channel set using the set identifier.

An event channel must have been created for this event channel set; otherwise, the request is rejected.

If an “Add SubUser Subscription” request is received to create a subscription and there is already an existing subscription with the same *subscriberId*, *targetId*, *targetSubId*, *event*, and *applicationId*, the second subscription is considered a duplicate of the first. Cisco BroadWorks then issues a *Subscription Terminated* event on the first subscription, and accepts the second subscription in place of the first one.

The expiration value must be within the range configured in the system. If the provided value exceeds the maximum allowed value, then the expiration value is set to the maximum allowed value. If the provided expiration value is less than 3600 (one hour), then the value is set to “3600”.

5.11.2.2 Request

A remote application can issue this request by sending an AddUserSubIdSubscriptionRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the subscription target.
<i>subId</i>	String	Yes	The subId of the target.
<i>subscription</i>	Subscription	Yes	The detail of the subscription to add.

Only the basic elements relevant for a new subscription should be provided by the subscription parameter. The following elements are ignored and should not be provided with this request:

- *subscriptionId*
- *subscriberId*
- *targetIdType*
- *targetGrpParentId*
- *targetId*
- *targetSubId*

5.11.2.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscription</i>	Subscription	No	Contains the detail of the created subscription. Only present when a successful response is returned (that is, the statusCode is set to "200").

5.11.2.4 Associated Events

The *Subscription Terminated* event can be received following a subscription creation.

An initial notification is sent for the subscription.

5.11.2.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
403	110842	Channel set <channelSetId> not found.
For <i>System</i> errors, see section 5.1.3.5 System Errors .		
For <i>Target User</i> errors, see section 5.1.3.1 Target User Errors .		
For <i>Subscription</i> errors, see section 5.1.3.6 Subscription Errors .		



5.11.3 Add Group Subscription (AS only)

5.11.3.1 Description

This request is used to add a subscription against the specified enterprise/service provider group on behalf of the authenticated subscriber in Cisco BroadWorks. This subscription is for an event package, and triggers notifications back to the issuing remote client application when those events are triggered. The subscription is associated with an event channel set using the set identifier.

An event channel must have been created for this event channel set; otherwise, the request is rejected.

If an "Add Group Subscription" request is received to create a subscription and there is already an existing subscription with the same *subscriberId*, *targetId*, *event*, and *applicationId*, the second subscription is considered a duplicate of the first. Cisco BroadWorks then issues a *Subscription Terminated* event on the first subscription, and accepts the second subscription in place of the first one.

The expiration value must be within the range configured in the system. If the provided value exceeds the maximum allowed value, then the expiration value is set to the maximum allowed value. If the provided expiration value is less than 3600 (one hour), then the value is set to "3600".

5.11.3.2 Request

A remote application can issue this request by sending an *AddGroupSubscriptionrequest*. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>serviceProviderId</i>	String	Yes	The enterprise ID or service provider ID of the group.
<i>groupId</i>	String	Yes	The group ID of the group.
<i>subscription</i>	Subscription	Yes	The detail of the subscription to add.

Only the basic elements relevant for a new subscription should be provided by the *subscription* parameter. The following elements are ignored and should not be provided with this request:

- *subscriptionId*
- *subscriberId*
- *targetIdType*
- *targetGrpParentId*
- *targetId*
- *targetSubId*

5.11.3.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscription</i>	Subscription	No	Contains the detail of the created subscription. Only present when a successful response is returned (that is, the statusCode is set to "200").

5.11.3.4 Associated Events

The *Subscription Terminated* event can be received following a subscription creation.

No initial notification is sent for the subscription.

5.11.3.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
403	110860	Group <groupId> not found.
403	110842	Channel set <channelSetId> not found.
For System errors, see section 5.1.3.5 System Errors .		
For Subscription errors, see section 5.1.3.6 Subscription Errors .		

5.11.4 Add Enterprise Subscription (AS only)

5.11.4.1 Description

This request is used to add a subscription against the specified enterprise on behalf of the authenticated subscriber in Cisco BroadWorks. This subscription is for an event package, and triggers notifications back to the issuing remote client application when those events are triggered. The subscription is associated with an event channel set using the set identifier.

An event channel must have been created for this event channel set; otherwise, the request is rejected.

If an “Add Enterprise Subscription” request is received to create a subscription and there is already an existing subscription with the same *subscriberId*, *targetId*, *event*, and *applicationId*, the second subscription is considered a duplicate of the first. Cisco BroadWorks then issues a *Subscription Terminated* event on the first subscription, and accepts the second subscription in place of the first one.

The expiration value must be within the range configured in the system. If the provided value exceeds the maximum allowed value, then the expiration value is set to the maximum allowed value. If the provided expiration value is less than 3600 (one hour), then the value is set to “3600”.

5.11.4.2 Request

A remote application can issue this request by sending an AddEnterpriseSubscriptionRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>enterpriseld</i>	String	Yes	The enterprise ID of the enterprise.
<i>subscription</i>	Subscription	Yes	The detail of the subscription to add.

NOTE: AddEnterpriseSubscriptionRequest and AddServiceProviderSubscriptionRequest are interchangeable. You can use the AddEnterpriseSubscriptionRequest to add a Service Provider Subscription by setting the enterpriseld to the Service Provider ID value in the request.

Only the basic elements relevant for a new subscription should be provided by the subscription parameter. The following elements are ignored and should not be provided with this request:

- subscriptionId
- subscriberId
- targetIdType
- targetGrpParentId
- targetId
- targetSubId

5.11.4.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscription</i>	Subscription	No	Contains the detail of the created subscription. Only present when a successful response is returned (that is, the statusCode is set to "200").

5.11.4.4 Associated Events

The *Subscription Terminated* event can be received following a subscription creation.

No initial notification is sent for the subscription.

5.11.4.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
403	110861	Service provider or enterprise <serviceProviderId or enterpriseld> not found.
403	110842	Channel set <channelSetId> not found.

For System errors, see section [5.1.3.5 System Errors](#).

statusCode	errorCode	Summary
------------	-----------	---------

For Subscription errors, see section 5.1.3.6 Subscription Errors .
--

5.11.5 Add Service Provider Subscription (**AS only**)

5.11.5.1 Description

This request is used to add a subscription against the specified service provider on behalf of the authenticated subscriber in Cisco BroadWorks. This subscription is for an event package, and triggers notifications back to the issuing remote client application when those events are triggered. The subscription is associated with an event channel set using the set identifier. An event channel must have been created for this event channel set; otherwise, the request is rejected.

If an “Add Service Provider Subscription” request is received to create a subscription and there is already an existing subscription with the same *subscriberId*, *targetId*, *event*, and *applicationId*, the second subscription is considered a duplicate of the first. Cisco BroadWorks then issues a *Subscription Terminated* event on the first subscription, and accepts the second subscription in place of the first one.

The expiration value must be within the range configured in the system. If the provided value exceeds the maximum allowed value, then the expiration value is set to the maximum allowed value. If the provided expiration value is less than 3600 (one hour), then the value is set to “3600”.

5.11.5.2 Request

A remote application can issue this request by sending an *AddServiceProviderSubscriptionRequest*. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>serviceProviderId</i>	String	Yes	The service provider ID of the service provider.
<i>subscription</i>	Subscription	Yes	The detail of the subscription to add.

NOTE: *AddServiceProviderSubscriptionRequest* and *AddEnterpriseSubscriptionRequest* and are interchangeable. So one can use the *AddServiceProviderSubscriptionRequest* to add an Enterprise Subscription by setting the *serviceProviderId* to the Enterprise Id value in the request.

Only the basic elements relevant for a new subscription should be provided by the *subscription* parameter. The following elements are ignored and should not be provided with this request:

- *subscriptionId*
- *subscriberId*
- *targetIdType*
- *targetGrpParentId*
- *targetId*
- *targetSubId*

5.11.5.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscription</i>	Subscription	No	Contains the detail of the created subscription. Only present when a successful response is returned (that is, the statusCode is set to "200").

5.11.5.4 Associated Events

The *Subscription Terminated* event can be received following a subscription creation.

No initial notification is sent for the subscription.

5.11.5.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
403	110861	Service provider or enterprise <serviceProviderId or enterpriseId> not found.
403	110842	Channel set <channelSetId> not found.
For System errors, see section 5.1.3.5 System Errors .		
For Subscription errors, see section 5.1.3.6 Subscription Errors .		

5.11.6 Add System Subscription (**AS only**)

5.11.6.1 Description

This request is used to add a subscription against the system on behalf of the authenticated subscriber in Cisco BroadWorks. This subscription is for an event package, and triggers notifications back to the issuing remote client application when those events are triggered. The subscription is associated with an event channel set using the set identifier. An event channel must have been created for this event channel set; otherwise, the request is rejected.

If an “Add System Subscription” request is received to create a subscription and there is already an existing subscription with the same *subscriberId*, *targetId*, *event*, and *applicationId*, the second subscription is considered a duplicate of the first. Cisco BroadWorks then issues a *Subscription Terminated* event on the first subscription, and accepts the second subscription in place of the first one.

The expiration value must be within the range configured in the system. If the provided value exceeds the maximum allowed value, then the expiration value is set to the maximum allowed value. If the provided expiration value is less than 3600 (one hour), then the value is set to “3600”.

5.11.6.2 Request

A remote application can issue this request by sending an AddSystemSubscriptionRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscription</i>	Subscription	Yes	The detail of the subscription to add.

5.11.6.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscription</i>	Subscription	No	Contains the detail of the created subscription. Only present when a successful response is returned (that is, the statusCode is set to "200").

Only the basic elements relevant for a new subscription should be provided by the subscription parameter. The following elements are ignored and should not be provided with this request:

- subscriptionId
- subscriberId
- targetIdType
- targetGrpParentId
- targetId
- targetSubId

5.11.6.4 Associated Events

The *Subscription Terminated* event can be received following a subscription creation.

No initial notification is sent for the subscription.

5.11.6.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
403	110842	Channel set <channelSetId> not found.
For System errors, see section 5.1.3.5 System Errors .		
For Subscription errors, see section 5.1.3.6 Subscription Errors .		

5.11.7 Update Subscription

5.11.7.1 Description

This request is used to modify a subscription against the specified subscriber. It is typically used to refresh the subscription prior to its expiration.

The expiration value must be within the range configured in the system. If the provided value exceeds the maximum allowed value, then the expiration value is set to the maximum allowed value. If the provided expiration value is less than 3600 (1 hour), then the value is set to "3600".

5.11.7.2 Request

A remote application can issue this request by sending an UpdateSubscriptionRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriptionId</i>	String	Yes	The subscription ID of the subscription to update.
<i>subscription</i>	Subscription	Yes	The detail of the subscription to update.

5.11.7.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscription</i>	Subscription	No	Contains the detail of the updated subscription. Only present when a successful response is returned (that is, statusCode is set to "200").

5.11.7.4 Associated Events

If the request specifies a refresh value, then the request is treated as a refresh. In this case, if the target of the subscription is a subscriber then an initial notification is sent for the subscription. If the target of the subscription is a collection of subscribers, for example, a group, service provider/enterprise, or system, then no initial notification is sent for the subscription.

5.11.7.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
403	110862	Subscription ID <subscriptionId> not found

For System errors, see section [5.1.3.5 System Errors](#).

5.11.8 Delete Subscription

5.11.8.1 Description

This request is used to delete a single subscription

5.11.8.2 Request

A remote application can issue this request by sending a DeleteSubscriptionRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscriptionId</i>	String	Yes	The subscription ID of the subscription to delete.

5.11.8.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.11.8.4 Associated Events

No event is associated with this request.

5.11.8.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
403	110862	Subscription ID <subscriptionId> not found.
For System errors, see section 5.1.3.5 System Errors .		

5.11.9 Get Single Subscription

5.11.9.1 Description

This request is used to request the details of a single subscription. If the request is accepted and a subscription is found, then the returned response contains the associated subscription.

5.11.9.2 Request

A remote application can issue this request by sending a GetSingleSubscriptionRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscriptionId</i>	SubscriptionId	Yes	The subscription ID of the subscription to retrieve.

5.11.9.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscription</i>	Subscription	No	The subscription found. Only present when a successful response is returned (that is, statusCode is set to "200") and a subscription was found with the subscriptionId specified in the request.

5.11.9.4 Associated Events

There is no event associated with this request.

5.11.9.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
403	110862	Subscription ID <subscriptionId> not found.
For System errors, see section 5.1.3.5 System Errors .		

5.11.10 Get Subscription

5.11.10.1 Description

This request is used to request the details of all the subscriptions that match the specified criteria in the request.

If no criteria are included, then all subscriptions within the scope of the authorized subscriber for the request may be returned.

If criteria are included, then only subscriptions that are both within the scope of the authorized subscriber for the request and match the specified criteria may be returned.

If the request is accepted, then the returned response contains the list of subscriptions matching the requested criteria sorted by the "expires" value (smallest expires value first).

5.11.10.2 Request

A remote application can issue this request by sending a GetSubscriptionRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID of the owner of the subscription.

Parameter	Parameter Type	Required	Description
<i>targetEntityType</i>	EntityType	No	The type (user, group, service provider, enterprise, or system) of the target of the subscriptions to retrieve. If not present, then all subscriptions owned by a user in the authorized user's scope are retrieved subject to any specified parameters.
<i>targetGrpParentId</i>	String	No	The enterprise or ServiceProviderId of the target group of subscriptions to retrieve. This is required only when <i>targetEntityType</i> is group. If not present, then all subscriptions owned by a user in the authorized user's scope are retrieved subject to any specified parameters.
<i>targetId</i>	String	No	If the <i>targetId</i> is specified, filters subscriptions against this subscriber.
<i>targetSubId</i>	String	No	If the <i>targetSubId</i> is specified, filters subscriptions against this subId.
<i>extAppId</i>	String	No	If the <i>extAppId</i> is specified, filters subscription against this applicationId.
<i>event</i>	string	No	If the event is specified, filters subscriptions against this event package

5.11.10.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>subscriptions</i>	SubscriptionList	No	The list of subscriptions. There can be more than one subscription in the response. Only present when a successful response is returned (that is, the statusCode is set to 200) and one or more subscription matched the requested criteria.

5.11.10.4 Associated Events

There is no event associated with this request.

5.11.10.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
403	110860	Group <groupId> not found.
403	110861	Service provider or enterprise <serviceProviderId or enterpriseId> not found.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		
For Subscription errors, see section 5.1.3.6 Subscription Errors .		

5.12 Application Controller Request (AS only)

5.12.1 Set Application Controller State

5.12.1.1 Description

The remote application sends this request to set the application controller state to either “ready” or “not ready”.

This request is typically sent at the end of system initialization once the remote application has synchronized all subscribed resources related to the application controller. In this case, the remote application sets the state to “ready”.

The application controller is associated to an event channel set through configuration. If the event channel set is empty, then Cisco BroadWorks rejects any attempt to set the state to “ready”. The remote application must create event channels for the associated event channel set before setting the state to “ready”.

This request can also be used to force the application controller state to “not ready” and force evaluation of a failure condition on resources that are linked to the application controller (for example, a route point).

5.12.1.2 Request

A remote application can issue this request by sending a Set ApplicationControllerState Request. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>applicationControllerId</i>	String	Yes	The <i>applicationControllerId</i> that identifies the application controller.
<i>applicationController</i>	ApplicationController	Yes	The <i>applicationControl</i> parameter contains the request state, as either “ready” or “not ready”.

5.12.1.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.12.1.4 Associated Events

There is no event associated with this request.

5.12.1.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
403	110830	Application controller <AppControllerId> not found.
403	110831	No channels for application controller <AppControllerId>.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.12.2 Get Application Controller State

5.12.2.1 Description

The remote application sends this request to retrieve the application controller state. If the request is accepted, then the returned response contains the state information.

5.12.2.2 Request

A remote application can issue this request by sending a *Get ApplicationController State Request*. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>applicationControllerId</i>	String	Yes	The <i>applicationControllerId</i> that identifies the application controller.

5.12.2.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>applicationController</i>	ApplicationController	Yes	The current application controller. Contains the <i>applicationController State</i> requested. Only present when a successful response is returned (that is, the statusCode is set to 200).

5.12.2.4 Associated Events

There is no event associated with this request.

5.12.2.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
403	110830	Application controller <applicationControlledId> not found.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.13 Voice Messaging (AS only)

5.13.1 Delete Voice Messaging Message Request

5.13.1.1 Description

The remote application sends this request to delete a voice message.

5.13.1.2 Request

A remote application can issue this request by sending a DeleteVoiceMessagingMessageRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID associated with the voice message.
<i>messageId</i>	String	Yes	The ID of the message to be deleted.

5.13.1.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.13.1.4 Associated Events

There is no event associated with this request.

5.13.1.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110001	Error communicating with voice message repository.
400	110003	Message <messageId> not found.
400	110801	User is not assigned the Voice Messaging User service.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.13.2 Get Voice Messaging Message Request

5.13.2.1 Description

The remote application sends this request to retrieve the full content of a voice message.

In the event that a second download request arrives for the same subscriber while another download is already in progress, an error is returned to the second caller.

5.13.2.2 Request

A remote application can issue this request by sending a GetVoiceMessagingMessageRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>subscriberId</i>	SubscriberId	Yes	The subscriber ID associated with the voice message.
<i>messageId</i>	String	Yes	The ID of the message to be retrieved.

5.13.2.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>voiceMessage</i>	VoiceMessage	No	The detail of the message. Only present when the statusCode is set to "200". For more information, see section 8 Type Definitions .

5.13.2.4 Associated Events

There is no event associated with this request.

5.13.2.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110001	Error communicating with voice message repository.
400	110002	Message download already in progress.
400	110003	Message <messageld> not found.
400	110801	User is not assigned the Voice Messaging User service.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

5.13.3 Get All Voice Messaging Messages Request

5.13.3.1 Description

The remote application sends this request to retrieve a list of detailed information regarding each of the subscriber's voice messages.



5.13.3.2 Request

A remote application can issue this request by sending a GetAllVoiceMessagingMessagesRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
subscriberId	SubscriberId	Yes	The subscriber ID associated with the voice message.

5.13.3.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
VoiceMessagingMessages	VoiceMessagingMessages	No	The detail information about each of the voice messaging messages. Only present when the statusCode is set to "200". For more information, see section 8 Type Definitions .

5.13.3.4 Associated Events

There is no event associated with this request.

5.13.3.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110001	Error communicating with voice message repository.
400	110801	User is not assigned the Voice Messaging User service.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.13.4 Mark All Voice Messaging Messages as Unread Request

5.13.4.1 Description

The remote application sends this request to mark all their messages as "new" that is, not "read" yet.

5.13.4.2 Request

A remote application can issue this request by sending a MarkAllVoiceMessagingMessagesAsUnreadRequest. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
subscriberId	SubscriberId	Yes	The subscriber ID associated with the voice messages.

5.13.4.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.13.4.4 Associated Events

There is no event associated with this request.

5.13.4.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110001	Error communicating with voice message repository.
400	110801	User is not assigned the Voice Messaging User service.
For System errors, see section 5.1.3.5 System Errors .		
For Target User errors, see section 5.1.3.1 Target User Errors .		

5.13.5 Mark All Voice Messaging Messages as Read Request

5.13.5.1 Description

The remote application sends this request to mark all their messages as “read” that is, not “new” yet.

5.13.5.2 Request

A remote application can issue this request by sending a `MarkAllVoiceMessagingMessagesAsReadRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID associated with the voice messages.

5.13.5.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.13.5.4 Associated Events

There is no event associated with this request.

5.13.5.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110001	Error communicating with voice message repository.
400	110801	User is not assigned the Voice Messaging User service.
For System errors, see section 5.1.3.5 System Errors .		

For Target User errors, see section [5.1.3.1 Target User Errors](#).

5.13.6 Mark Voice Messaging Message as Unread Request

5.13.6.1 Description

The remote application sends this request to mark a specific message as “new” that is, not “read” yet.

5.13.6.2 Request

A remote application can issue this request by sending a `MarkVoiceMessagingMessageAsUnreadRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID associated with the voice message.
<code>messageId</code>	String	Yes	The ID of the message to be marked.

5.13.6.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.13.6.4 Associated Events

There is no event associated with this request.

5.13.6.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110001	Error communicating with voice message repository.
400	110002	Message download already in progress.
400	110003	Message <messageId> not found.
400	110801	User is not assigned the Voice Messaging User service.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).



5.13.7 Mark Voice Messaging Messages As Read Request

5.13.7.1 Description

The remote application sends this request to mark a specific message as “read” that is, not “new”.

5.13.7.2 Request

A remote application can issue this request by sending a `MarkVoiceMessagingMessagesAsReadRequest`. This request contains the common parameters listed in section [5.1.1 Request](#) and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>subscriberId</code>	SubscriberId	Yes	The subscriber ID associated with the voice message.
<code>messageId</code>	String	Yes	The ID of the message to be marked.

5.13.7.3 Response

The response contains only the common parameters listed in section [5.1.2 Response](#).

5.13.7.4 Associated Events

There is no event associated with this request.

5.13.7.5 Error Cases

The following table lists the specific errors associated with this request.

statusCode	errorCode	Summary
400	110001	Error communicating with voice message repository.
400	110002	Message download already in progress.
400	110003	Message <messageId> not found.
400	110801	User is not assigned the Voice Messaging User service.

For System errors, see section [5.1.3.5 System Errors](#).

For Target User errors, see section [5.1.3.1 Target User Errors](#).

6 Events

This section describes the various types of events sent by Cisco BroadWorks. It covers the following topics:

- Event types
- Call events
- Conference events
- Route Point events (**AS only**)
- ACD events (**AS only**)
- Agent events (**AS only**)
- ACD Configuration events (**AS only**)
- Call Park events (**AS only**)
- Meet-Me Conference events (**AS only**)
- Collaborate Room (**AS only**)
- Service Management events

6.1 Event and Event Response

For each event sent by Cisco BroadWorks, the remote application must send an event response. The details of the event and event response are listed in the table in the following subsection.

NOTE: The ChannelTerminatedEvent does not follow the event format described in this section, and an event response is not expected by Cisco BroadWorks. For more information regarding this event, see section [6.15.1 Channel Terminated](#).

6.1.1 Event Format

While there are several event types that can be generated by Cisco BroadWorks, all events share a common set of parameters. The following table lists these common parameters.

Parameter	Parameter Type	Required	Description
<i>requestId</i>	RequestId	Yes	An identification supplied by Cisco BroadWorks to map the event notification to the client response at the connection layer. Must be returned in the event response.
<i>eventId</i>	EventId	Yes	An identification supplied by Cisco BroadWorks to map the event notification to the client response at the application layer. Must be returned in the event response.
<i>sequence Number</i>	PositiveInteger	Yes	The sequence number for the notification.
<i>userId</i>	UserId	Yes	The userId of the user that owns the subscription for which the event is being sent.

Parameter	Parameter Type	Required	Description
<i>externalApplicationId</i>	ExternalApplicationId	Yes	The external application ID of the subscription for which the event is being sent.
<i>channelId</i>	ChannelId	No	The channelId identifies the channel used. Only present when using contact event channel for notification. Always present when using the CTI interface.
<i>subscriptionId</i>	SubscriptionId	Yes	The ID of the subscription for which the event is being sent.
<i>sharedSubscriptionId</i>	SubscriptionId	No	The ID of the shared subscription for which this event has been generated. It is present only if the event is generated as part of a subscription that contained a sharedSubscriptionId.
<i>targetId</i>	UserId	No	The ID of the target user for which the subscription applies; only present if the targetId is different from the user ID.
<i>targetSubId</i>	String	No	Used by specific event packages to provide a second level of granularity to the targetId. For example, a targetUserId for a Meet-Me Conference bridge can have a targetSubId, which identifies a specific conference that belongs to the bridge. Alternatively, a targetUserId for a Collaborate Room owner can have a targetSubId, which identifies a specific room that belongs to the owner.

Each specific event type contains these common parameters followed by some event specific parameters.

6.1.2 Event Response

A remote application must return an EventResponse to every event received. The following type definition table lists the parameters of an EventResponse.

Parameter	Parameter Type	Required	Description
<i>requestId</i>	RequestId	Yes	The request ID of the corresponding event.
<i>eventId</i>	EventId	Yes	The event ID of the corresponding event.
<i>statusCode</i>	EventResponseStatus Code	Yes	The response code. The remote application must set this value to "200" when the event is successfully accepted.
<i>reason</i>	String	Yes	The response reason. The remote application can set this value to any relevant value. This value can be used for troubleshooting purposes.

6.2 Call Events

A call event is an event that reports a change in the state of a call or of a call session. Each call event typically maintains one or multiple parameters of type Call. While a Call is a generic type and can carry a lot of information, not all available parameters are used by each specific event. Within each event description, there is a table that shows the parameters that are used.

The next subsections provide detailed information for all the available call event types:

- Call Originating
- Call Originated
- Call Received
- Call Answered
- Call Held
- Call Retrieved
- Call Updated
- Call Redirected
- Call Transferred
- Call Releasing
- Call Released
- Call Collecting
- Call Picked-up
- Call Barged-in
- Call Monitored (AS only)
- Call Forwarded
- Call Detached
- Call Subscription
- Call Subscription Resync (AS only)
- CallRecordingStartingEvent (AS only)
- CallRecordingStartedEvent (AS only)
- CallRecordingPausedEvent (AS only)
- CallRecordingResumedEvent (AS only)
- CallRecordingStoppedEvent (AS only)
- CallSecurityClassificationUpdatedEvent (AS only)
- CallClientSessionInfoUpdated (AS only)
- HookStatus (AS only)

6.2.1 Call Originating

The *Call Originating* event is generated when a dial request is accepted by Cisco BroadWorks and an attempt is made to alert the subscriber's endpoint(s). The call ID and external tracking ID are generated at this point and included in the event. The call state is *Alerting* and the personality is Click-to-Dial. The address parameter of the remoteParty is set to the dialed phone number.

6.2.1.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `CallOriginatingEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>call</code>	Call	Yes	Contains the details of a call. See the call type definitions in section 8 Type Definitions for more information.

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<code>callId</code>	Mandatory	
<code>extTrackingId</code>	Mandatory	
<code>personality</code>	Mandatory	
<code>state</code>	Mandatory	
<code>appearance</code>	Optional	Only present when a specific appearance is assigned to the call.
<code>remoteParty</code>	Mandatory	
<code>startTime</code>	Mandatory	
<code>clientSessionInfo</code>	Optional	Only present if client session info has been set for this session.

6.2.2 Call Originated

The *Call Originated* event is generated when a call is being originated by the subscriber. This can happen in two cases:

- When the calling subscriber manually initiates a call by dialing a destination on its device. In this case, a new call ID and the external tracking ID are generated by Cisco BroadWorks.
- When the remote application has initiated a call through a dial request. In this case, first a *Call Originating* event is generated when the system has accepted the request. Once the subscriber has accepted the dial attempt (by going off-hook), then the call personality changes from "Click-to-Dial" to "Originator", and a *Call Originated* event is issued. Both the *Call Originating* and *Call Originated* event have the same call ID and external-tracking-id.

In both cases, the call state is *Alerting* and the personality is originator.

6.2.2.1 Event Type Definition

Cisco BroadWorks issues this event by sending a CallOriginatedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For more information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>endpoint</i>	Mandatory	
<i>appearance</i>	Optional	Only present when a specific appearance is assigned to the call.
<i>startTime</i>	Mandatory	
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>allowedRecordingControls</i>	Optional	Only present when the user has the call recording service active.
<i>recordingState</i>	Optional	Only present when the user has the call recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is not an active recording.
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.3 Call Received

A *CallReceivedEvent* is generated when a subscriber receives an incoming call. When subscribed to the *Advanced Call* event package, the event is generated before service processing, for example, for Call Forwarding Always (CFA). When subscribed to the *Standard Call* or *Basic Call* event packages, the event is generated after service processing, when the subscriber's device receives the incoming call. This event is also generated when a subscriber is being recalled (for example, for Call Transfer Recall). It contains information about the caller in the *remoteParty* element. The call state is "Alerting" and the personality is "Terminator".



6.2.3.1 Event Type Definition

Cisco BroadWorks issues this event by sending a CallReceivedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>networkCallId</i>	Optional	Only present when the remote party is a network party
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>redirections</i>	Optional	Present if a prior redirection took place.
<i>alternateAddressType</i>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.
<i>acdCallInfo</i>	Optional	Present when the call is delivered by an ACD or Route Point.
<i>huntGroupUserId</i>	Optional	Present when the call is an unanswered hunt group termination.
<i>distinctiveRing</i>	Optional	Present for incoming calls for which a distinctive ring pattern should apply.
<i>diversionInhibited</i>	Optional	Present when the diversions are inhibited for the call.
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an "Automatic Hold/Retrieve" recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.

Parameter	Mandatory or Optional	Comments
<code>executiveUserId</code>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<code>recallFromParty</code>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<code>clientSessionInfo</code>	Optional	Only present if client session info has been set for this session.

6.2.4 Call Answered

The *Call Answered* event is generated when a call originated or received by the subscriber is answered.

For a given call, there may be more than one *Call Answered* event generated. This can happen for instance if an Active call is redirected following a Blind Transfer request. In such case, the call will go back in the *Alerting* state. Once the transferred destination answers the call, a new *Call Answered* event is generated. As a result, a total of two *Call Answered* events will have been received: The first one before the transfer, (since the call was already in the active state prior to the transfer) and the second one after the successful transfer.

The answerTime captured in the event always indicates the time of the first call answer. The call state is *Active*.

6.2.4.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `CallAnsweredEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>call</code>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<code>callId</code>	Mandatory	
<code>extTrackingId</code>	Mandatory	
<code>networkCallId</code>	Optional	Only present when the remote party is a network party.
<code>personality</code>	Mandatory	
<code>state</code>	Mandatory	
<code>remoteParty</code>	Mandatory	
<code>endpoint</code>	Optional	Present when an endpoint is associated with the call.
<code>appearance</code>	Optional	Present only when a specific appearance is assigned to the call.
<code>alternateAddressType</code>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.

Parameter	Mandatory or Optional	Comments
<i>acdCallInfo</i>	Optional	Present when the call is delivered by an ACD or Route Point.
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an "Automatic Hold/Retrieve" recall by Cisco BroadWorks)
<i>startTime</i>	Mandatory	
<i>answerTime</i>	Mandatory	
<i>totalHoldTime</i>	Optional	Only present if the call had been answered and was at least once in the <i>Held</i> state.
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls</i>	Optional	Only present when the user has the Call Recording service active.
<i>recordingState</i>	Optional	Only present when the user has the Call Recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is no active recording.
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<i>securityClassification</i>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.5 Call Held

The *Call Held* event is generated when the subscriber or the remote party holds the call. The event indicates whether the call was held by the subscriber or remote party. The call state is *Held* or *Remote Held* depending on which party held the call. Remote party call hold is only reported for group/enterprise calls in a standalone application. In addition, if the call is already held locally, then the remote party call hold is not reported.

6.2.5.1 Event Type Definition

Cisco BroadWorks issues this event by sending a CallHeldEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>networkCallId</i>	Optional	Only present when the remote party is a network party.
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>endpoint</i>	Optional	Present when an endpoint is associated with the call.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the call.
<i>alternateAddressType</i>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.
<i>acdCallInfo</i>	Optional	Present when the call is delivered by an ACD or Route Point.
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an "Automatic Hold/Retrieve" recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	
<i>answerTime</i>	Mandatory	
<i>holdTime</i>	Mandatory	
<i>totalHoldTime</i>	Optional	Only present if the call was at least once in the <i>Held</i> state.
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>recorded</i>	Optional	Only present if the call is being recorded.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls</i>	Optional	Only present when the user has the Call Recording service active.

Parameter	Mandatory or Optional	Comments
<i>recordingState</i>	Optional	Only present when the user has the Call Recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is no active recording.
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<i>securityClassification</i>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.6 Call Retrieved

The *Call Retrieved* event is generated when the subscriber or remote party retrieves a held call. The event indicates whether the call was retrieved by the subscriber or remote party. The call state is *Active*. The remote party call retrieve is only reported for group/enterprise calls in a standalone application. In addition, if the call is already held locally, then the remote party call retrieved is not reported.

6.2.6.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *CallRetrievedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>networkCallId</i>	Optional	Only present when the remote party is a network party.
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>endpoint</i>	Optional	Present when an endpoint is associated with the call.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the call.

Parameter	Mandatory or Optional	Comments
<i>alternateAddressType</i>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.
<i>acdCallInfo</i>	Optional	Present when the call is delivered by an ACD or Route Point.
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an "Automatic Hold/Retrieve" recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	
<i>answerTime</i>	Mandatory	
<i>totalHoldTime</i>	Optional	Only present if the call was at least once in the <i>Held</i> state.
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>recorded</i>	Optional	Only present if the call is being recorded.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls</i>	Optional	Only present when the user has the Call Recording service active.
<i>recordingState</i>	Optional	Only present when the user has the Call Recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is no active recording.
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<i>securityClassification</i>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.7 Call Updated

The *Call Updated* event is generated when call attributes are modified, with the call state not being modified. Examples of such updates are: remote party information changes, local endpoint changes, personality changes. Local endpoint changes occurs if for instance a subscriber has multiple access endpoints and performs a Call Retrieve from a phone, then the AccessEndpoint is expected to be modified. A Personality change can occur during a Directed Call Pickup with Barge-in.

As a result, a remote application must inspect the various field of the *Call Updated* event to determine what change occurred since the last notification.



6.2.7.1 Event Type Definition

Cisco BroadWorks issues this event by sending a CallUpdatedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>networkCallId</i>	Optional	Only present when the remote party is a network party.
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>redirections</i>	Optional	Present if a prior redirection took place.
<i>endpoint</i>	Optional	Present when an endpoint is associated with the call.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the call.
<i>allowAnswer</i>	Optional	Present when the endpoint supports protocol extensions that allow an application to answer a call in the <i>Alerting</i> state with a Terminator personality or Click-to-Dial personality using a talk request.
<i>allowRetrieve</i>	Optional	Present when the endpoint supports protocol extensions to retrieve a held call without requiring a subscriber action.
<i>alternateAddressType</i>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.
<i>acdCallInfo</i>	Optional	Only present when the call is delivered by an ACD or Route Point.
<i>huntGroupUserId</i>	Optional	Present when the call is an unanswered hunt group termination.
<i>distinctiveRing</i>	Optional	Present for incoming calls for which a distinctive ring pattern should apply.
<i>diversionInhibited</i>	Optional	Present when the diversions are inhibited for the call.
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an "Automatic Hold/Retrieve" recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	

Parameter	Mandatory or Optional	Comments
<i>answerTime</i>	Optional	Present if the call was answered.
<i>totalHoldTime</i>	Optional	Only present if the call was at least once in the <i>Held</i> state.
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>groupCallParkLocationSelected</i>	Optional	Only present when the Group Call Park feature has been invoked and the parked location selected. This indicates that the <i>remoteParty</i> element contains the parked location.
<i>recorded</i>	Optional	Only present if the call is being recorded.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls</i>	Optional	Only present when the user has the Call Recording service active.
<i>recordingState</i>	Optional	Only present when the user has the Call Recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is no active recording.
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<i>securityClassification</i>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.8 Call Redirected

The *Call Redirected* event is generated when one or more calls are redirected by the subscriber. The event includes the redirection reason which indicates the type of redirection (for example, simultaneous ring, call forward always, blind transfer). The call state is set to “Detached”.

6.2.8.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *CallRedirectedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>callList</i>	CallList	Yes	Contains a list of calls. For information on the call type definitions, see section 8 Type Definitions .



As shown, this event can contain more than one call parameter. This can happen for instance when performing a Consultative Transfer. In a Consultative Transfer, two calls go simultaneously into the *Detached* state. This event will list the details of the two calls involved in the redirection.

The following table provides the list of parameters used for each call parameter listed in the callList.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>networkCallId</i>	Optional	Only present when the remote party is a network party.
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>redirect</i>	Mandatory	
<i>endpoint</i>	Optional	Present when an endpoint is associated with the call.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the call.
<i>alternateAddressType</i>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.
<i>acdCallInfo</i>	Optional	Only present when the call is delivered by an ACD or Route Point.
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an "Automatic Hold/Retrieve" recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	
<i>detachedTime</i>	Mandatory	
<i>answerTime</i>	Optional	Present if the call was answered.
<i>totalHoldTime</i>	Optional	Only present if the call was at least once in the <i>Held</i> state.
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>recorded</i>	Optional	Only present if the call was being recorded.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls</i>	Optional	Only present when the user has the Call Recording service active.

Parameter	Mandatory or Optional	Comments
<i>recordingState</i>	Optional	Only present when the user has the Call Recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is no active recording.
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<i>securityClassification</i>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.9 Call Transferred

The *Call Transferred* event is generated when the call is transferred by the remote party, resulting in a call state or external tracking ID change. This event is only generated for group/enterprise call redirections. The external-tracking-id and personality are modified in the case of a transfer with consultation.

6.2.9.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *CallTransferredEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>networkCallId</i>	Optional	Only present when the remote party is a network party.
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>acdCallInfo</i>	Optional	Only present when the call is delivered by an ACD or Route Point.
<i>endpoint</i>	Optional	Present when an endpoint is associated with the call.

Parameter	Mandatory or Optional	Comments
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the call.
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an "Automatic Hold/Retrieve" recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	
<i>answerTime</i>	Optional	Present if the call was answered.
<i>totalHoldTime</i>	Optional	Only present if the call was at least once in the <i>Held</i> state.
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>recorded</i>	Optional	Only present if the call is being recorded.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls</i>	Optional	Only present when the user has the Call Recording service active.
<i>recordingState</i>	Optional	Only present when the user has the Call Recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is no active recording.
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<i>securityClassification</i>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.10 Call Releasing

The Call Releasing event is generated when a call is connecting to a treatment. The call is eventually released when the treatment finishes playing or when the subscriber hangs up. The release cause is set to the appropriate value. The address parameter of the remoteParty is set to the dialed digits.

6.2.10.1 Event Type Definition

Cisco BroadWorks issues this event by sending a CallReleasingEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>releaseCause</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>endpoint</i>	Optional	Present when an endpoint is associated with the call.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the call.
<i>startTime</i>	Mandatory	
<i>answerTime</i>	Optional	Present if the call was answered.
<i>detachedTime</i>	Optional	Present if the call was in Detached state.
<i>releaseTime</i>	Mandatory	
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>recorded</i>	Optional	Only present if the call was being recorded.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls</i>	Optional	Only present when the user has the Call Recording service active.
<i>recordingState</i>	Optional	Only present when the user has the Call Recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is no active recording.

Parameter	Mandatory or Optional	Comments
<code>executiveUserId</code>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<code>securityClassification</code>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined.
<code>clientSessionInfo</code>	Optional	Only present if client session info has been set for this session.

6.2.11 Call Released

The *Call Released* event is generated when a call is released, either by the subscriber or the remote party. The event is also generated when the dial request is aborted without the subscriber going off-hook. The call state is released and the release cause is set.

6.2.11.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `CallReleasedEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>call</code>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<code>callId</code>	Mandatory	
<code>extTrackingId</code>	Mandatory	
<code>networkCallId</code>	Optional	Only present when the remote party is a network party.
<code>personality</code>	Mandatory	
<code>state</code>	Mandatory	
<code>releaseCause</code>	Optional	Present only if the call is being released due to a failure of some kind.
<code>remoteParty</code>	Mandatory	
<code>acdCallInfo</code>	Optional	Only present when the call is delivered by an ACD or Route Point.
<code>redirect</code>	Optional	Present if the call was redirected.
<code>endpoint</code>	Optional	Present when an endpoint is associated with the call.
<code>appearance</code>	Optional	Present only when a specific appearance is assigned to the call.
<code>alternateAddressType</code>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.

Parameter	Mandatory or Optional	Comments
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an “Automatic Hold/Retrieve” recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	
<i>answerTime</i>	Optional	Present if the call was answered.
<i>detachedTime</i>	Option	Present if the call was in the <i>Detached</i> state prior to the release.
<i>releaseTime</i>	Mandatory	
<i>releasingParty</i>	Mandatory	
<i>totalHoldTime</i>	Optional	Only present if the call was at least once in the <i>Held</i> state.
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>recorded</i>	Optional	Only present if the call was being recorded.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls</i>	Optional	Only present when the user has the Call Recording service active.
<i>recordingState</i>	Optional	Only present when the user has the Call Recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is no active recording.
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<i>securityClassification</i>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.12 Call Collecting

The *Call Collecting* event is generated when Cisco BroadWorks needs to collect information by interacting with a subscriber. This event generated in two types of situation:

- When programming a service through a feature access code. This happens for instance when a subscriber activates Call Forward Always.

- When Cisco BroadWorks requires some information before calling the remote party. This happens for instance when a subscriber invoke the Calling Line ID Blocking service before dialing. This can also happen when Cisco BroadWorks require an authentication or accounting code before calling the remote party. Note that in these cases, the *Call Collecting* event is not generated if the service activation precedes the setup of a call and the destination is already known from the dialed digits.

The call ID and external tracking ID is generated when the call is established with Cisco BroadWorks and included in the event. The call state is *Active* and the personality is Originator. The address parameter of the remoteParty is set to the dialed digits.

6.2.12.1 Event Type Definition

Cisco BroadWorks issues this event by sending a CallCollectingEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>endpoint</i>	Mandatory	
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the call.
<i>startTime</i>	Mandatory	
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.13 Call Picked-up

The *Call Picked-up* event is generated when a call is picked-up as a result of a Directed Call Pickup request or Directed Call Pickup with Barge-in where the call state was in *Alerting*. The event is generated against the subscriber for which the call is picked-up. The new external tracking ID is reported in this event. The call state is *Alerting*.

6.2.13.1 Event Type Definition

BroadWorks issues this event by sending a CallPickedUpEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>networkCallId</i>	Optional	Only present when the remote party is a network party.
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>endpoint</i>	Optional	Present when an endpoint is associated with the call.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the call.
<i>alternateAddressType</i>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an "Automatic Hold/Retrieve" recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	
<i>answerTime</i>	Mandatory	
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>recorded</i>	Optional	Only present if the call is being recorded.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls</i>	Optional	Only present when the user has the Call Recording service active.
<i>recordingState</i>	Optional	Only present when the user has the Call Recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is no active recording.

Parameter	Mandatory or Optional	Comments
<code>executiveUserId</code>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<code>recallFromParty</code>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<code>securityClassification</code>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined.
<code>clientSessionInfo</code>	Optional	Only present if client session info has been set for this session.

6.2.14 Call Barged-in

The *Call Barged-in* event is generated when a call is conference in as a result of a Directed Call Pickup with Barge-in request. The event is generated against the subscriber for which the call is barged-in. The new external tracking ID is reported in this event. The call state is either *Active* or *Held*.

6.2.14.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `CallBargedInEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>call</code>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<code>callId</code>	Mandatory	
<code>extTrackingId</code>	Mandatory	
<code>networkCallId</code>	Optional	Only present when the remote party is a network party.
<code>personality</code>	Mandatory	
<code>state</code>	Mandatory	
<code>remoteParty</code>	Mandatory	
<code>endpoint</code>	Optional	Present when an endpoint is associated with the call.
<code>appearance</code>	Optional	Present only when a specific appearance is assigned to the call.
<code>alternateAddressType</code>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.
<code>acdCallInfo</code>	Optional	Only present when the call is delivered by an ACD or Route Point.

Parameter	Mandatory or Optional	Comments
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an “Automatic Hold/Retrieve” recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	
<i>answerTime</i>	Mandatory	
<i>totalHoldTime</i>	Optional	Only present if the call was at least once in the <i>Held</i> state.
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>recorded</i>	Optional	Only present if the call is being recorded.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls</i>	Optional	Only present when the user has the Call Recording service active.
<i>recordingState</i>	Optional	Only present when the user has the Call Recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is no active recording.
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<i>securityClassification</i>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.15 Call Monitored (AS only)

The *Call Monitored* event is generated when a call is monitored as a result of a Monitor Call or Monitor Next Call request. The new external tracking ID is reported in this event. The call state is *Active*.

6.2.15.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *CallMonitoredEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .



The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>endpoint</i>	Optional	Present when an endpoint is associated with the call.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the call.
<i>alternateAddressType</i>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.
<i>acdCallInfo</i>	Optional	Only present when the call is delivered by an ACD or Route Point.
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an "Automatic Hold/Retrieve" recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	
<i>answerTime</i>	Mandatory	
<i>totalHoldTime</i>	Optional	Only present if the call was at least once in the <i>Held</i> state.
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>recorded</i>	Optional	Only present if the call is being recorded.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls</i>	Optional	Only present when the user has the Call Recording service active.
<i>recordingState</i>	Optional	Only present when the user has the Call Recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is not an active recording.
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.16 Call Forwarded

The *Call Forwarded* event is generated when a call is forwarded as a result of the application of a Cisco BroadWorks service (for example, Call Forwarding Always). The call state is *Alerting*. The personality is Originator. The *remoteParty* value depends on the service assigned to the subscriber as follows:

- If the subscriber has the Connected Line Identification Presentation assigned, then the *remoteParty* specifies where the call was redirected.
- If the subscriber does not have the Connected Line Identification Presentation assigned, then the *remoteParty* specifies the destination initially dialed.

6.2.16.1 Event Type Definition

Cisco BroadWorks issues this event by sending a CallForwardedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>networkCallId</i>	Optional	Only present when the remote party is a network party.
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>endpoint</i>	Optional	Present when an endpoint is associated with the call.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the call.
<i>startTime</i>	Mandatory	
<i>answerTime</i>	Optional	Present if the call was answered.
<i>totalHoldTime</i>	Optional	Only present if the call was at least once in the <i>Held</i> state.
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>recorded</i>	Optional	Only present if the call is being recorded.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.

Parameter	Mandatory or Optional	Comments
<i>allowedRecordingControls</i>	Optional	Only present when the user has the Call Recording service active.
<i>recordingState</i>	Optional	Only present when the user has the Call Recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is not an active recording.
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.17 Call Detached

The *Call Detached* event is generated when a call enters the detached state without being redirected. The call state is set to "Detached". This event is only sent for the Advanced Call event package when the personality is "Terminator" and for unanswered calls.

6.2.17.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *CallDetachedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>networkCallId</i>	Optional	Only present when the remote party is a network party.
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>alternateAddressType</i>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.
<i>acdCallInfo</i>	Optional	Present when the call is delivered by an ACD or Route Point.
<i>huntGroupUserId</i>	Optional	Present when the call is an unanswered hunt group termination.

Parameter	Mandatory or Optional	Comments
<i>distinctiveRing</i>	Optional	Present for incoming calls for which a distinctive ring pattern should apply.
<i>diversionInhibited</i>	Optional	Present when the diversions are inhibited for the call.
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an "Automatic Hold/Retrieve" recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.18 Call Park Retrieved

The Call Park Retrieved event is generated when a parked call is retrieved. The event is generated against the subscriber for which the call was parked and now retrieved. The new external tracking ID is reported in this event.

6.2.18.1 Event Type Definition

Cisco BroadWorks issues this event by sending a CallParkRetrievedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For more information, see section 8 Type Definitions .



The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<code>callId</code>	Mandatory	
<code>extTrackingId</code>	Mandatory	
<code>networkCallId</code>	Optional	It is only present when the remote party is a network party.
<code>personality</code>	Mandatory	
<code>state</code>	Mandatory	
<code>remoteParty</code>	Mandatory	
<code>endpoint</code>	Optional	It is present when an endpoint is associated with the call.
<code>appearance</code>	Optional	It is present only when a specific appearance is assigned to the call.
<code>alternateAddressType</code>	Optional	It is present when an alternate address is in use. It is not present when the subscriber's primary address is in use.
<code>acdCallInfo</code>	Optional	It is only present when the call is delivered by an ACD or Route Point.
<code>recallType</code>	Optional	It is present when the call is a recall (for example, following an Automatic Hold/Retrieve recall by Cisco BroadWorks).
<code>startTime</code>	Mandatory	
<code>answerTime</code>	Mandatory	
<code>totalHoldTime</code>	Optional	It is only present if the call was at least once in the <i>Held</i> state.
<code>agentEscalation</code>	Optional	It is only present when the call is in relation to an agent escalation.
<code>acdOutgoingCallInfo</code>	Optional	It is only present when an agent makes an outgoing call as an ACD or Route Point.
<code>recorded</code>	Optional	Only present if the call was being recorded.
<code>findMeFollowMeUserId</code>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<code>allowedRecordingControls</code>	Optional	Only present when the user has the Call Recording service active.
<code>recordingState</code>	Optional	Only present when the user has the Call Recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is not an active recording.
<code>executiveUserId</code>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.

Parameter	Mandatory or Optional	Comments
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<i>securityClassification</i>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.19 Call Recording Starting (AS only)

The *Call Recording Starting Event* is generated when the call recording request has been processed but the recording cannot be started yet because the call is held.

6.2.19.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *CallRecordingStartingEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>networkCallId</i>	Optional	Only present when the remote party is a network party.
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>redirections</i>	Optional	Present if a prior redirection took place.
<i>endpoint</i>	Optional	Present when an endpoint is associated with the call.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the call.
<i>allowAnswer</i>	Optional	Present when the endpoint supports protocol extensions that allow an application to answer a call in the <i>Alerting</i> state with a Terminator personality or Click-to-Dial personality using a talk request.
<i>allowRetrieve</i>	Optional	Present when the endpoint supports protocol extensions to retrieve a held call without requiring a subscriber action.



Parameter	Mandatory or Optional	Comments
<i>alternateAddressType</i>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.
<i>acdCallInfo</i>	Optional	Only present when the call is delivered by an ACD or Route Point.
<i>huntGroupUserId</i>	Optional	Present when the call is an unanswered hunt group termination.
<i>distinctiveRing</i>	Optional	Present for incoming calls for which a distinctive ring pattern should apply.
<i>diversionInhibited</i>	Optional	Present when the diversions are inhibited for the call.
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an "Automatic Hold/Retrieve" recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	
<i>answerTime</i>	Optional	Present if the call was answered.
<i>totalHoldTime</i>	Optional	Only present if the call was at least once in the <i>Held</i> state.
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>recorded</i>	Mandatory	Indicates that the call is being recorded.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls</i>	Mandatory	The recording controls that are allowed for this call. Set to stop if the recording mode allows the user to stop the recording
<i>recordingState</i>	Mandatory	Set to "Pending".
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<i>securityClassification</i>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.20 Call Recording Started (AS only)

The *Call Recording Started* event is generated when call recording has started successfully.

6.2.20.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *CallRecordingStartedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>networkCallId</i>	Optional	Only present when the remote party is a network party.
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>redirections</i>	Optional	Present if a prior redirection took place.
<i>endpoint</i>	Optional	Present when an endpoint is associated with the call.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the call.
<i>allowAnswer</i>	Optional	Present when the endpoint supports protocol extensions that allow an application to answer a call in the <i>Alerting</i> state with a Terminator personality or Click-to-Dial personality using a talk request.
<i>allowRetrieve</i>	Optional	Present when the endpoint supports protocol extensions to retrieve a held call without requiring a subscriber action.
<i>alternateAddressType</i>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.
<i>acdCallInfo</i>	Optional	Only present when the call is delivered by an ACD or Route Point.
<i>huntGroupUserId</i>	Optional	Present when the call is an unanswered hunt group termination.
<i>distinctiveRing</i>	Optional	Present for incoming calls for which a distinctive ring pattern should apply.
<i>diversionInhibited</i>	Optional	Present when the diversions are inhibited for the call.

Parameter	Mandatory or Optional	Comments
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an "Automatic Hold/Retrieve" recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	
<i>answerTime</i>	Optional	Present if the call was answered.
<i>totalHoldTime</i>	Optional	Only present if the call was at least once in the <i>Held</i> state.
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>recorded</i>	Mandatory	Indicates that the call is being recorded.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls</i>	Mandatory	The recording controls that are allowed for this call. Set based on the user's recording mode and current recording state.
<i>recordingState</i>	Mandatory	Set to "Started".
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<i>securityClassification</i>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.21 Call Recording Paused (AS only)

The *Call Recording Paused* event is generated when call recording is paused successfully.

6.2.21.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *CallRecordingPausedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>networkCallId</i>	Optional	Only present when the remote party is a network party.
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>redirections</i>	Optional	Present if a prior redirection took place.
<i>endpoint</i>	Optional	Present when an endpoint is associated with the call.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the call.
<i>allowAnswer</i>	Optional	Present when the endpoint supports protocol extensions that allow an application to answer a call in the <i>Alerting</i> state with a Terminator personality or Click-to-Dial personality using a talk request.
<i>allowRetrieve</i>	Optional	Present when the endpoint supports protocol extensions to retrieve a held call without requiring a subscriber action.
<i>alternateAddressType</i>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.
<i>acdCallInfo</i>	Optional	Only present when the call is delivered by an ACD or Route Point.
<i>huntGroupUserId</i>	Optional	Present when the call is an unanswered hunt group termination.
<i>distinctiveRing</i>	Optional	Present for incoming calls for which a distinctive ring pattern should apply.
<i>diversionInhibited</i>	Optional	Present when the diversions are inhibited for the call.
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an "Automatic Hold/Retrieve" recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	
<i>answerTime</i>	Optional	Present if the call was answered.
<i>totalHoldTime</i>	Optional	Only present if the call was at least once in the <i>Held</i> state.
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>recorded</i>	Mandatory	Indicates that the call is being recorded.

Parameter	Mandatory or Optional	Comments
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls</i>	Mandatory	The recording controls that are allowed for this call. Set based on the user's recording mode and current recording state.
<i>recordingState</i>	Mandatory	Set to "Paused".
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<i>securityClassification</i>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.22 Call Recording Resumed (AS only)

The *Call Recording Resumed* event is generated when call recording is resumed successfully.

6.2.22.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *CallRecordingResumedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>networkCallId</i>	Optional	Only present when the remote party is a network party.
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>redirections</i>	Optional	Present if a prior redirection took place.
<i>endpoint</i>	Optional	Present when an endpoint is associated with the call.



Parameter	Mandatory or Optional	Comments
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the call.
<i>allowAnswer</i>	Optional	Present when the endpoint supports protocol extensions that allow an application to answer a call in the <i>Alerting</i> state with a Terminator personality or Click-to-Dial personality using a talk request.
<i>allowRetrieve</i>	Optional	Present when the endpoint supports protocol extensions to retrieve a held call without requiring a subscriber action.
<i>alternateAddressType</i>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.
<i>acdCallInfo</i>	Optional	Only present when the call is delivered by an ACD or Route Point.
<i>huntGroupUserId</i>	Optional	Present when the call is an unanswered hunt group termination.
<i>distinctiveRing</i>	Optional	Present for incoming calls for which a distinctive ring pattern should apply.
<i>diversionInhibited</i>	Optional	Present when the diversions are inhibited for the call.
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an "Automatic Hold/Retrieve" recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	
<i>answerTime</i>	Optional	Present if the call was answered.
<i>totalHoldTime</i>	Optional	Only present if the call was at least once in the <i>Held</i> state.
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>recorded</i>	Mandatory	Indicates that the call is being recorded.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls</i>	Mandatory	The recording controls that are allowed for this call. Set based on the user's recording mode and current recording state.
<i>recordingState</i>	Mandatory	Set to "Started".
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.

Parameter	Mandatory or Optional	Comments
<i>securityClassification</i>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.23 Call Recording Stopped (AS only)

The *Call Recording Stopped* event is generated when call recording is stopped because the user sent an explicit stop recording request or because the call recording failed.

6.2.23.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *CallRecordingResumedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .
<i>reason</i>	RecordingStoppedReason	Yes	The reason for why the recording has been stopped.

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>networkCallId</i>	Optional	Only present when the remote party is a network party.
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>redirections</i>	Optional	Present if a prior redirection took place.
<i>endpoint</i>	Optional	Present when an endpoint is associated with the call.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the call.
<i>allowAnswer</i>	Optional	Present when the endpoint supports protocol extensions that allow an application to answer a call in the <i>Alerting</i> state with a Terminator personality or Click-to-Dial personality using a talk request.
<i>allowRetrieve</i>	Optional	Present when the endpoint supports protocol extensions to retrieve a held call without requiring a subscriber action.

Parameter	Mandatory or Optional	Comments
<i>alternateAddressType</i>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.
<i>acdCallInfo</i>	Optional	Only present when the call is delivered by an ACD or Route Point.
<i>huntGroupUserId</i>	Optional	Present when the call is an unanswered hunt group termination.
<i>distinctiveRing</i>	Optional	Present for incoming calls for which a distinctive ring pattern should apply.
<i>diversionInhibited</i>	Optional	Present when the diversions are inhibited for the call.
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an "Automatic Hold/Retrieve" recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	
<i>answerTime</i>	Optional	Present if the call was answered.
<i>totalHeldTime</i>	Optional	Only present if the call was at least once in the <i>Held</i> state.
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>recorded</i>	Mandatory	Indicates that the call is being recorded.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls</i>	Mandatory	The recording controls that are allowed for this call. Set based on the user's recording mode and current recording state.
<i>recordingState</i>	Optional	Only present if the recording was stopped due to a failure.
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<i>securityClassification</i>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined.
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

If the recording was a successful user-initiated stop recording request, the *reason* element is set to "User Requested". Users who can stop their recording are also allowed to start a new recording. Since a new recording can be started, the *allowedRecordingControls* element is set to "record" and the *recordingState* element is absent, which indicates that there is no active recording. Even though the recording has been stopped, the *recorded* element is present to indicate that the record command was requested at least once for this call.



If the recording was stopped due to a failure, the *reason* element is set to “Failure”, the *recordingState* element is set to “Failed”, and the *allowedRecordingControls* element is set to “none”. Even though the recording has been stopped, the *recorded* element is still present to indicate that the record command was requested at least once for this call.

6.2.24 Call Subscription and Call Subscription Resync (AS only)

The *Call Subscription* event is issued immediately after a subscription is added or refreshed, provided that the subscription target is not a collection of subscriber (for example, a group) and that contact was registered with the subscription.

The *CallSubscriptionResyncEvent* is produced whenever the Application Server shuts down abnormally allowing a remote application to resynchronize.

Both events share the same definition.

6.2.24.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *CallSubscriptionEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>callList</i>	CallList	No	Contains the list of calls maintained by the subscriber. For more information, see section 8 Type Definitions . Only present if there is at least one call maintained by the subscriber.
<i>conference</i>	Conference	No	The details of the conference maintained by the target subscriber. This parameter is only present if the subscriber is a conference controller.
<i>hookStatus</i>	HookStatus	Yes	Indicates the hook status of the user.

The following table provides the list of parameters used by this event for each call parameter.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>networkCallId</i>	Optional	Only present when the remote party is a network party.
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>redirections</i>	Optional	Only present for unanswered, incoming calls.
<i>redirect</i>	Optional	Only present when the subscriber has redirected the call.
<i>endpoint</i>	Optional	Only present when an endpoint is associated with the call.
<i>appearance</i>	Optional	Only present when the call has an appearance value assigned.



Parameter	Mandatory or Optional	Comments
<code>allowAnswer</code>	Optional	Present when the endpoint supports protocol extensions that allow an application to answer a call in the <i>Alerting</i> state with a Terminator personality or Click-to-Dial personality using a talk request.
<code>allowRetrieve</code>	Optional	Only present when the protocol extensions are supported by the subscriber endpoint.
<code>alternateAddressType</code>	Optional	Only present when an alternate address is in use.
<code>acdCallInfo</code>	Optional	Only present when the call is delivered by an ACD or Route Point.
<code>huntGroupUserId</code>	Optional	Only present when the call is delivered by an agent.
<code>distinctiveRing</code>	Optional	Only present for incoming calls for which a distinctive ring pattern should apply.
<code>diversionInhibited</code>	Optional	Only present when diversions are inhibited for the call.
<code>recallType</code>	Optional	Only present if the call was triggered by a callback or recall tracked by RecallType.
<code>startTime</code>	Mandatory	
<code>answerTime</code>	Optional	Only present if the call was answered.
<code>heldTime</code>	Optional	Only present if the call is in the <i>Held</i> state.
<code>totalHeldTime</code>	Optional	Only present if the call was at least once in the <i>Held</i> state.
<code>agentEscalation</code>	Optional	Only present when the call is in relation to an agent escalation.
<code>acdOutgoingCallInfo</code>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<code>recorded</code>	Mandatory	Only present when the call is being recorded.
<code>findMeFollowMeUserId</code>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<code>allowedRecordingControls</code>	Optional	Only present when the user has the Call Recording service active.
<code>recordingState</code>	Optional	Only present when the user has the Call Recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is no active recording.
<code>executiveUserId</code>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<code>recallFromParty</code>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<code>securityClassification</code>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined.

Parameter	Mandatory or Optional	Comments
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

The following table provides the list of parameters used by this event for each conference parameter.

Parameter	Mandatory or Optional	Comments
<i>state</i>	Mandatory	
<i>endPoint</i>	Optional	Present when an endpoint is associated with the conference.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the conference.
<i>conferenceType</i>	Optional	Only present when the conference was established as a result of a barge-in or silent monitoring operation.
<i>mute</i>	Optional	Only present when the conference controller is muted (for example, as a result of a silent monitoring operation).
<i>conferenceParticipantList</i>	Mandatory	

The following table lists the parameter used by this event for the hookStatus parameter.

Parameter	Mandatory or Optional	Comments
<i>hookStatus</i>	Mandatory	

6.2.25 Call Security Classification Updated (**AS only**)

The *Call Security Classification Updated* event is generated when there are changes to the security classification characteristics of a call for users who have the Security Classification service assigned.

The event contains the name of the call classification level.

The event is generated when:

- There is a change in the call classification level caused by a user changing their user classification level.
- There is a change in the call classification level caused by the remote user changing their user classification level.
- There are changes to the parties connected in the call as in complex call scenarios (that is, transfers) and conference calls.



6.2.25.1 Event Type Definition

Cisco BroadWorks issues this event by sending a CallSecurityClassificationUpdatedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>networkCallId</i>	Optional	Only present when the remote party is a network party.
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>endpoint</i>	Optional	Present when an endpoint is associated with the call.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the call.
<i>alternateAddressType</i>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.
<i>acdCallInfo</i>	Optional	Present when the call is delivered by an ACD or Route Point.
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an "Automatic Hold/Retrieve" recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	
<i>answerTime</i>	Mandatory	
<i>heldTime</i>	Mandatory	
<i>totalHeldTime</i>	Optional	Only present if the call was at least once in the <i>Held</i> state.
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>recorded</i>	Optional	Only present if the call is being recorded.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.

Parameter	Mandatory or Optional	Comments
<i>allowedRecordingControls</i>	Optional	Only present when the user has the Call Recording service active.
<i>recordingState</i>	Optional	Only present when the user has the Call Recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is not an active recording.
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<i>securityClassification</i>	Mandatory	
<i>clientSessionInfo</i>	Optional	Only present if client session info has been set for this session.

6.2.26 Call Client Session Info Updated (AS only)

The *Call Client Session Info Updated* event is generated when there are changes to the client session info for the call.

6.2.26.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *CallClientSessionInfoUpdatedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>call</i>	Call	Yes	Contains the details of a call. For information on the call type definitions, see section 8 Type Definitions .

The following table provides the list of call parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>networkCallId</i>	Optional	Only present when the remote party is a network party.
<i>personality</i>	Mandatory	
<i>state</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>endpoint</i>	Optional	Present when an endpoint is associated with the call.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the call.

Parameter	Mandatory or Optional	Comments
<i>alternateAddressType</i>	Optional	Present when an alternate address is in use. It is not present when the subscriber's primary address is in use.
<i>acdCallInfo</i>	Optional	Present when the call is delivered by an ACD or Route Point.
<i>recallType</i>	Optional	Present when the call is a recall (for example, following an "Automatic Hold/Retrieve" recall by Cisco BroadWorks).
<i>startTime</i>	Mandatory	
<i>answerTime</i>	Mandatory	
<i>heldTime</i>	Mandatory	
<i>totalHeldTime</i>	Optional	Only present if the call was at least once in the <i>Held</i> state.
<i>agentEscalation</i>	Optional	Only present when the call is in relation to an agent escalation.
<i>acdOutgoingCallInfo</i>	Optional	Only present when an agent makes an outgoing call as an ACD or Route Point.
<i>recorded</i>	Optional	Only present if the call is being recorded.
<i>findMeFollowMeUserId</i>	Optional	Only present when the call was redirected to a user by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls</i>	Optional	Only present when the user has the Call Recording service active.
<i>recordingState</i>	Optional	Only present when the user has the Call Recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is not an active recording.
<i>executiveUserId</i>	Optional	Only present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty</i>	Optional	Only present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.
<i>securityClassification</i>	Optional	Only present when the Security Classification service is assigned to the user and the call classification level has been determined
<i>clientSessionInfo</i>	Mandatory	

6.2.27 Hook Status (**AS only**)

The *Hook Status* event is issued application when a subscriber:

- Originates a call.
- Answers a call.
- Releases a call while not being active on any other call.
- Performs a line seize.

- Places a call using an MGCP device.

The event indicates whether the user is “Off-Hook” or “On-Hook”.

6.2.27.1 Event Type Definition

Cisco BroadWorks issues this event by sending a HookStatusEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>hookStatus</i>	HookStatus	Yes	Indicates the hook status of a user.

6.3 Conference Events

A conference is an event that reports a change in the state of a conference.

Each conference event maintains Conference parameters. While a Conference is a generic type and can carry a lot of information, not all available parameters are used by each specific event. Within each event description, there is a table that shows the parameters that are used. For more information, see section [8 Type Definitions](#).

The next subsections provide detailed information for all the available conference event types:

- Conference Started
- Conference Held
- Conference Retrieved
- Conference Updated
- Conference Released
- Conference Call Muted
- Conference Call Unmuted
- Conference Call Made Deaf
- Conference Call Made Undeaf

6.3.1 Conference Started

The *Conference Started* event is produced when a conference is started. The conference state is *Active*.

This event provides the details of the conference including the list of conference participant. Depending on the conference type, Endpoint-initiated conference, or Cisco BroadWorks conference, this event can list zero or more participants. For more information on conference type, see section [4.2.5 Minimum Number of Participants in Conference](#).

For an endpoint-initiated conference, the *Conference Started* event does not list any participants. The list of participants is provided later through *Conference Updated* event as participants are added to the conference. For Cisco BroadWorks conference, the *Conference Started* event lists all participants (two or more) present in the initial *ConferenceStartRequest*.

The *Conference Started* event reports the conference state as *Active*.

6.3.1.1 Event Type Definition

Cisco BroadWorks issues this event by sending a ConferenceStartedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>conference</i>	Conference	Yes	The detail of the conferences. For more information, see section 8 Type Definitions .

The following table provides the list of conference parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>state</i>	Mandatory	
<i>endPoint</i>	Optional	Present when an endpoint is associated with the conference.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the conference.
<i>conferenceType</i>	Optional	Only present when the conference was established as a result of a barge-in or silent monitoring operation.
<i>muted</i>	Optional	Only present when the conference controller is muted (for example, as a result of a silent monitoring operation).
<i>conferenceParticipantList</i>	Optional	Present when there are participants in the conference. Depending on the conference type, the conference can contain zero or more participants. For more information, see section 4.2.5 Minimum Number of Participants in Conference .

6.3.2 Conference Held

The *Conference Held* event is produced when the conference is put on hold. The conference state is *Held*.

6.3.2.1 Event Type Definition

Cisco BroadWorks issues this event by sending a ConferenceHeldEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>conference</i>	Conference	Yes	The detail of the conferences. For more information, see section 8 Type Definitions .

The following table provides the list of conference parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>state</i>	Mandatory	
<i>endPoint</i>	Optional	Present when an endpoint is associated with the conference.

Parameter	Mandatory or Optional	Comments
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the conference.
<i>conferenceType</i>	Optional	Only present when the conference was established as a result of a barge-in or silent monitoring operation.
<i>conferenceParticipantList</i>	Mandatory	
<i>muted</i>	Optional	Only present when the conference controller is muted (for example, as a result of a silent monitoring operation).

6.3.3 Conference Retrieved

The *Conference Retrieved* event is produced when the conference is retrieved from hold. The conference state is *Active*.

6.3.3.1 Event Type Definition

Cisco BroadWorks issues this event by sending a ConferenceRetrievedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>conference</i>	Conference	Yes	The detail of the conferences. For more information, see section 8 Type Definitions .

The following table provides the list of conference parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>state</i>	Mandatory	
<i>endPoint</i>	Optional	Present when an endpoint is associated with the conference.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the conference.
<i>conferenceType</i>	Optional	Only present when the conference was established as a result of a barge-in or silent monitoring operation.
<i>conferenceParticipantList</i>	Mandatory	
<i>muted</i>	Optional	Only present when the conference controller is muted (for example, as a result of a silent monitoring operation).

6.3.4 Conference Updated

The *Conference Updated* event is produced when the conference parameters are modified, with the conference state not being modified. Examples of such updates are: local endpoint changes, participant added, and participant removed. Local endpoint changes can occur if a subscriber performs a Call Retrieve from a phone, then the *AccessEndpoint* is expected to be modified.

6.3.4.1 Event Type Definition

Cisco BroadWorks issues this event by sending a ConferenceUpdatedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>conference</i>	Conference	Yes	The detail of the conferences. For more information, see section 8 Type Definitions .

The following table provides the list of conference parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>state</i>	Mandatory	
<i>endPoint</i>	Optional	Present when an endpoint is associated with the conference.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the conference.
<i>conferenceType</i>	Optional	Only present when the conference was established as a result of a barge-in or silent monitoring operation.
<i>conferenceParticipantList</i>	Mandatory	
<i>muted</i>	Optional	Only present when the conference controller is muted (for example, as a result of a silent monitoring operation).

6.3.5 Conference Released

The *Conference Released* event is produced when the conference is released. The conference state is released.

6.3.5.1 Event Type Definition

Cisco BroadWorks issues this event by sending a ConferenceReleasedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>conference</i>	Conference	Yes	The detail of the conferences. For more information, see section 8 Type Definitions .

The following table provides the conference parameter used by this event.

Parameter	Mandatory or Optional	Comments
<i>state</i>	Mandatory	

6.3.6 Conference Muted

The *Conference Muted* event is produced when the conference controller is muted.

6.3.6.1 Event Type Definition

Cisco BroadWorks issues this event by sending a ConferenceMutedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>conference</i>	Conference	Yes	The detail of the conference. For more information, see section 8 Type Definitions .

The following table provides the list of conference parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>state</i>	Mandatory	
<i>endPoint</i>	Optional	Present when an endpoint is associated with the conference.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the conference.
<i>conferenceType</i>	Optional	Only present when the conference was established as a result of a barge-in or silent monitoring operation.
<i>conferenceParticipantList</i>	Mandatory	
<i>muted</i>	Optional	Only present when the conference controller is muted (for example, as a result of a silent monitoring operation).

6.3.7 Conference Unmuted

The *Conference Unmuted* event is produced when the conference controller is unmuted.

6.3.7.1 Event Type Definition

Cisco BroadWorks issues this event by sending a ConferenceUnmutedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>conference</i>	Conference	Yes	The detail of the conference. For more information, see section 8 Type Definitions .

The following table provides the list of conference parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>state</i>	Mandatory	
<i>endPoint</i>	Optional	Present when an endpoint is associated with the conference.

Parameter	Mandatory or Optional	Comments
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the conference.
<i>conferenceType</i>	Optional	Only present when the conference was established as a result of a barge-in or silent monitoring operation.
<i>conferenceParticipantList</i>	Mandatory	
<i>muted</i>	Optional	Only present when the conference controller is muted (for example, as a result of a silent monitoring operation).

6.3.8 Conference Call Muted

The *Conference Call Muted* event is produced when a call is muted through a Conference Mute Call request. The conferenceParticipant in the conferenceParticipantList specifies which call is muted or not.

6.3.8.1 Event Type Definition

Cisco BroadWorks issues this event by sending a ConferenceCallMutedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>conference</i>	Conference	Yes	The detail of the conferences. For more information, see section 8 Type Definitions .

The following table provides the list of conference parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>state</i>	Mandatory	
<i>endPoint</i>	Optional	Present when an endpoint is associated with the conference.
<i>appearance</i>	Optional	Only present when a specific appearance is assigned to the conference.
<i>conferenceType</i>	Optional	Only present when the conference was established as a result of a barge-in or silent monitoring operation.
<i>conferenceParticipantList</i>	Mandatory	
<i>muted</i>	Optional	Only present when the conference controller is muted (for example, as a result of a silent monitoring operation).

6.3.9 Conference Call Unmuted

The *Conference Call Unmuted* event is produced when a call is no longer muted through a Conference UnMute Call request. The conferenceParticipant in the conferenceParticipantList specifies if a call is muted or not.

6.3.9.1 Event Type Definition

Cisco BroadWorks issues this event by sending a ConferenceRetrievedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
conference	Conference	Yes	The detail of the conferences. For more information, see section 8 Type Definitions .

The following table provides the list of conference parameters used by this event.

Parameter	Mandatory or Optional	Comments
state	Mandatory	
endPoint	Optional	Present when an endpoint is associated with the conference.
appearance	Optional	Only present when a specific appearance is assigned to the conference.
conferenceType	Optional	Only present when the conference was established as a result of a barge-in or silent monitoring operation.
conferenceParticipantList	Mandatory	
muted	Optional	Only present when the conference controller is muted (for example, as a result of a silent monitoring operation).

6.3.10 Conference Call Made Deaf

The *Conference Call Made Deaf* event is produced when a call is made deaf through a Conference Deaf Call request. The conferenceParticipant in the conferenceParticipantList specifies which call is deaf or not.

6.3.10.1 Event Type Definition

Cisco BroadWorks issues this event by sending a ConferenceCallMadeDeafEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
conference	Conference	Yes	The detail of the conferences. For more information, see section 8 Type Definitions .

The following table provides the list of conference parameters used by this event.

Parameter	Mandatory or Optional	Comments
state	Mandatory	
endPoint	Optional	Present when an endpoint is associated with the conference.
appearance	Optional	Present only when a specific appearance is assigned to the conference.

Parameter	Mandatory or Optional	Comments
<i>conferenceType</i>	Optional	Only present when the conference was established as a result of a barge-in or silent monitoring operation.
<i>conferenceParticipantList</i>	Mandatory	
<i>muted</i>	Optional	Only present when the conference controller is muted (for example, as a result of a silent monitoring operation).

6.3.11 Conference Call Made Undeaf

The *Conference Call Made Undeaf* event is produced when a call is no longer deaf through a Conference Undeaf call. The *conferenceParticipant* in the *conferenceParticipantList* specifies if a call is deaf or not.

6.3.11.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *ConferenceCallMadeUndeafEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>conference</i>	Conference	Yes	The detail of the conferences. For more information, see section 8 Type Definitions .

The following table provides the list of conference parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>state</i>	Mandatory	
<i>endPoint</i>	Optional	Present when an endpoint is associated with the conference.
<i>appearance</i>	Optional	Present only when a specific appearance is assigned to the conference.
<i>conferenceType</i>	Optional	Only present when the conference was established as a result of a barge-in or silent monitoring operation.
<i>conferenceParticipantList</i>	Mandatory	
<i>muted</i>	Optional	Only present when the conference controller is muted (for example, as a result of a silent monitoring operation).

6.4 Route Point Events (AS only)

A Route Point event is an event that reports a change associated to a call maintained in the Route Point queue.

The following subsections provide detailed information for all the available Route Point event types:

- Route Point Call Added
- Route Point Call Offered to Agent
- Route Point Call Answered by Agent

- Route Point Call Abandoned
- Route Point Call Overflowed
- Route Point Call Overflowed Treatment Completed
- Route Point Call Transferred
- Route Point Call Updated
- Route Point Call Bounced
- Route Point Call Forwarded
- Route Point Call Forwarded Treatment Completed
- Route Point Call Released
- Route Point Call Failed
- Route Point Outgoing Call Originated
- Route Point Outgoing Call Answered
- Route Point Treatment Started
- Route Point Treatment Completed
- Route Point MOH Started
- Route Point MOH Completed
- Route Point Ringback Started
- Route Point Ringback Completed
- Route Point Busy Started
- Route Point Busy Completed
- Route Point Silence Started
- Route Point Silence Completed
- Route Point Whisper Started
- Route Point Holiday Policy Applied
- Route Point Holiday Policy Treatment Completed
- Route Point Night Policy Applied
- Route Point Night Policy Treatment Completed
- Route Point Failed
- Route Point Recovered
- Route Point Subscription
- Route Point Subscription Resync

6.4.1 Route Point Call Added

The *Route Point Call Added* event is produced when a new call is added to the Route Point queue. For information on incoming call processing for details of what makes a call eligible to be added to the queue, see section [4.3.1 Incoming Call Processing](#).

6.4.1.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointCallAddedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The detail of the call queued at the Route Point. For more information, see section 8 Type Definitions .
<i>redirections</i>	RedirectionList	No	Present if a prior redirection took place. This parameter lists all the redirections of the received call. These are the details of other parties that have redirected this call prior to reaching the route point.

The following table provides the list of queueEntry parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.

6.4.2 Route Point Call Offered to Agent

The *Route Point Call Offered* event is produced when the remote application performs a Route Point Distribute Call against a call queued to an agent. This event indicates that the call is offered to the agent but remains in the queue.

6.4.2.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointCallOfferedToAgentEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The detail of the call queued at the Route Point. For more information, see section 8 Type Definitions .

The following table provides the list of queueEntry parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	

Parameter	Mandatory or Optional	Comments
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>playCollectInfo</i>	Optional	Present when a play-collect operation is performed while call is offered to the agent.
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

6.4.3 Route Point Call Answered by Agent

The *Route Point Call Answered* event is produced after a call has been offered to an agent, the agent has decided to accept the call, and the call has been removed from the queue. The *removeTime* parameter is set to the time the call was removed from the queue (that is, answered by an agent).

Note that when a whisper is played to the agent, then this event is only generated at the end of the whisper message.

6.4.3.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *RoutePointCallAnsweredByAgentEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	<i>RoutePointQueueEntry</i>	Yes	The detail of the call queued at the Route Point. For more information, see section 8 Type Definitions .

The following table provides the list of *queueEntry* parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>removeTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	

Parameter	Mandatory or Optional	Comments
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.
<i>answeringUserId</i>	Optional	Present when the agent that answered the call is a Cisco BroadWorks subscriber.
<i>answeringCallId</i>	Optional	Present when the call offer from the call center to the agent does not contain a SIP call leg.
<i>answeringNetworkCallId</i>	Optional	Present when the call offer from the call center to the agent contains a SIP call leg.

6.4.4 Route Point Call Abandoned

The *Route Point Call Abandoned* event is produced after a call is released by the remote party while being in the Route Point queue.

6.4.4.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `RoutePointCallAbandoned`. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	<code>RoutePointQueueEntry</code>	Yes	The detail of the call queued at the Route Point. For more information, see section 8 Type Definitions .

The following table provides the list of `queueEntry` parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>removeTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

6.4.5 Route Point Call Overflowed

The *Route Point Call Overflowed* event is produced after a call is marked as overflow. The *overFlowReason* specifies the cause of overflow.

6.4.5.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *RoutePointCallOverflowedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

The following table provides the list of parameters used.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	<i>RoutePointQueueEntry</i>	Yes	The details of the <i>RoutePointQueueEntry</i> . A <i>RoutePointQueueEntry</i> contains information about a call in the <i>RoutePoint</i> queue. For more information, see section 8 Type Definitions .
<i>overFlowReason</i>	<i>OverFlowReason</i>	No	Specify the reason for overflow. For more information, see section 8 Type Definitions .
<i>redirect</i>	<i>Redirect</i>	No	Present if the call was redirected to another destination based on call overflowed policy. This happens when the policy action is set to "transfer". For more information, see section 8 Type Definitions .
<i>redirections</i>	<i>RedirectionList</i>	No	Present if a prior redirection took place. This parameter lists all the redirections of the received call. These are the details of other parties that have redirected this call prior to reaching the route point. The parameter is not present when the call is overflowed because it stayed in the queue for too long (time overflow). This is because this event is issued when a call is already inside the queue and redirection information has already been provided by the <i>Route Point Call Added</i> event.
<i>treatmentStarted</i>	<i>EmptyContent</i>	No	Present if the policy starts playing an announcement or ringback according to the policy configuration.

The following table provides the list of *queueEntry* parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>removeTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	

Parameter	Mandatory or Optional	Comments
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

6.4.6 Route Point Call Overflowed Treatment Completed

The *Route Point Call Overflowed Treatment Completed* event is produced when the announcement is completed for a call treated according to the Overflow policy.

6.4.6.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *RoutePointCallOverflowedTreatment CompletedEvent*. This event contains the common parameters listed at the beginning of this section.

The following table provides the list of parameters used.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The details of the <i>RoutePointQueueEntry</i> . A <i>RoutePointQueueEntry</i> contains information about a call in the <i>RoutePoint</i> queue. For more information, see section 8 Type Definitions .
<i>overFlowReason</i>	OverFlowReason	No	Specify the reason for overflow. For more information, see section 8 Type Definitions .
<i>redirect</i>	Redirect	No	Present if the call was redirected to another destination based on call overflowed policy. This happens when the policy action is set to "transfer". For more information, see section 8 Type Definitions .

The following table provides the list of *queueEntry* parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>removeTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

6.4.7 Route Point Call Transferred

The *Route Point Call Transferred* event is produced when the remote application performs a blind transfer against a call queued in a Route Point.

6.4.7.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointCallTransferredEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The detail of the call queued at the Route Point. For more information, see section 8 Type Definitions .
<i>redirect</i>	Redirect	Yes	The detail of the redirect. For more information, see section 8 Type Definitions .

The following table provides the list of queueEntry parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>removeTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>Redirect</i>	Mandatory	
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

6.4.8 Route Point Call Updated

The *Route Point Call Updated* event is generated in any of the following cases:

- When the external tracking ID is changed, (this can happen for instance following a consultation transfer).
- When the CPD result is available for an outgoing dial.
- When the remote party information is modified.

6.4.8.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointCallUpdatedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The detail of the call queued at the Route Point. For more information, see section 8 Type Definitions .
<i>cpdResult</i>	CPDResult	No	Specify the Call Progress Detection result for the call. Present for outgoing call where CPD information is available.
<i>reason</i>	CallUpdateReason	No	Specify the reason for the update. Only present when the external tracking ID or the remote party information is changed.

The following table provides the list of queueEntry parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>playCollectInfo</i>	Optional	Present when a play-collect operation is currently performed.
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

6.4.9 Route Point Call Bounced

The *Route Point Call Bounced* event is produced after a call is bounced. When a call is bounced, the Route Point bounced call policy is applied as described in section [4.3.1.1 Bounced Call Policy](#). The event reports that the call is bounced by adding the *bounced* parameter to the RoutePointQueueEntry in the event. This parameter will be maintained in all future events until the call is removed from the Route Point queue.

6.4.9.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointCallBouncedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The detail of the call queued at the Route Point. For more information, see section 8 Type Definitions .

The following table provides the list of queueEntry parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Mandatory	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>playCollectInfo</i>	Optional	Present when a play-collect operation is currently performed.
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

6.4.10 Route Point Call Forwarded

The *Route Point Call Forwarded* event is produced when a call is forwarded following the application of the Forced Forwarding policy.

6.4.10.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointCallForwardedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>callId</i>	CallId	Yes	The call ID of the call.
<i>extTrackingId</i>	ExternalTrackingId	Yes	The external tracking ID of the call.
<i>routePointName</i>	String	Yes	Indicates the Route Point name (DNIS Name) associated with the RP number the call was received on.
<i>routePointNumber</i>	Address	Yes	Indicates the Route Point number (DNIS) the call was received on.
<i>remoteParty</i>	PartyInformation	Yes	Information about the remote party.

Parameter	Parameter Type	Required	Description
<i>redirect</i>	Redirect	Yes	The detail of the redirect.
<i>redirections</i>	RedirectionList	No	Present if a prior redirection took place. This parameter lists all the redirections of the received call. These are the details of other parties that have redirected this call prior to reaching the Route Point.
<i>treatmentStarted</i>	EmptyContent	No	Present if the policy starts playing an announcement or ringback according to the policy configuration.

6.4.11 Route Point Call Forwarded Treatment Completed

The Route Point Call Forwarded Treatment Completed event is produced when the announcement is completed for a call treated according to the Forced Forwarding policy.

6.4.11.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *RoutePointCallForwardedTreatmentCompletedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>callId</i>	CallId	Yes	The call ID of the call.
<i>extTrackingId</i>	ExternalTrackingId	Yes	The external tracking ID of the call.
<i>routePointName</i>	String	Yes	Indicates the Route Point name (DNIS Name) associated with the Route Point number the call was received on.
<i>routePointNumber</i>	Address	Yes	Indicates the Route Point number (DNIS) the call was received on.
<i>remoteParty</i>	PartyInformation	Yes	Information about the remote party.
<i>redirect</i>	Redirect	Yes	The detail of the redirect.

6.4.12 Route Point Call Released

The Route Point Call Released event is produced in two cases:

- Following a Route Point Release Call request.
- When an outgoing call is released during call setup, in which case the CPD result is provided (if available).

The removeTime is set to the time the call was released.



6.4.12.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointCallReleasedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The detail of the call queued at the Route Point. For more information, see section 8 Type Definitions .
<i>releaseReason</i>	RoutePointCallReleaseReason	Yes	The reason for the call release.
<i>outgoingCallReleaseCause</i>	ReleaseCause	No	Present if an outgoing call is released with cause (for example, called party is busy).
<i>cpdResult</i>	CPDResult	No	Specifies the Call Progress Detection result for the call. Present for outgoing call where CPD information is available.

The following table provides the list of queueEntry parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>removeTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

6.4.13 Route Point Call Failed

The *Route Point Call Failed* event is produced in two cases:

- When the Route Point per call failure policy is triggered. This can happen for instance if no request is received for a call within the configurable time period after being added to the queue.
- When the Route Point failure policy is triggered. This can happen for instance when a remote application sends a Route Point Fail request and a call is queued at the Route Point. This can also occur when a call is received for a failed Route Point.

6.4.13.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointCallFailedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The detail of the call queued at the Route Point. For more information, see section 8 Type Definitions .
<i>failureReason</i>	RoutePointCallFailureReason	Yes	The detail of the failure. For more information, see section 8 Type Definitions .

The following table provides the list of queueEntry parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>removeTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

6.4.14 Route Point Outgoing Call Originated

The *Route Point Call Originated* event is produced when a call is being originated following an Outgoing Dial action but before the call is added to the Route Point queue. The *outgoingCall* parameter is present to mark the call permanently as an outbound call.

6.4.14.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointCallOriginatedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The detail of the call queued at the Route Point. For more information, see section 8 Type Definitions .



The following table provides the list of queueEntry parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>outgoingCall</i>	Mandatory	

6.4.15 Route Point Outgoing Call Answered

The *Route Point Call Answered* event is produced when a call is answered by the remote destination. The *OutgoingCallAnswerTime* is set to the time the call was answered.

6.4.15.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *RoutePointOutgoingCallAnsweredEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	<i>RoutePointQueueEntry</i>	Yes	The detail of the call queued at the Route Point. For more information, see section 8 Type Definitions .

The following table provides the list of queueEntry parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>outgoingCall</i>	Mandatory	
<i>outgoingCallAnswerTime</i>	Mandatory	

6.4.16 Route Point Subscription Event and Route Point Subscription Resync Event

The *Route Point Subscription* event is issued immediately after a subscription is added or refreshed, provided that the subscription target is not a collection of subscribers (for example, a group) and that contacts were registered with the subscription.

The *Route Point Subscription Resync* event is issued whenever the Application Server shuts down abnormally allowing a remote application to resynchronize.

Both the *Route Point Subscription* event and the *Route Point Subscription Resync* event share the same definition.

6.4.16.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `RoutePointSubscriptionEvent` (or a `RoutePointSubscriptionResyncEvent`). This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>queueEntries</code>	<code>RoutePointQueueEntryList</code>	No	Contains the list of <code>RoutePointQueueEntry</code> maintained by the Route Point. For information, see section 8 Type Definitions . Only present if there is at least one <code>RoutePointQueueEntry</code> maintained by the Route Point.

The following table provides the list of `queueEntry` parameters used by each `RoutePointQueueEntry` in the list.

Parameter	Mandatory or Optional	Comments
<code>callId</code>	Mandatory	
<code>extTrackingId</code>	Mandatory	
<code>remoteParty</code>	Mandatory	
<code>addTime</code>	Mandatory	
<code>bounced</code>	Optional	Present only if call was bounced.
<code>routePointNumber</code>	Mandatory	
<code>routePointName</code>	Mandatory	
<code>playCollectInfo</code>	Optional	Present when a play-collect operation is currently being performed.
<code>outgoingCall</code>	Optional	Present only for outgoing call.
<code>outgoingCallAnswerTime</code>	Optional	Present only if outgoing call was answered.
<code>answeringUserId</code>	Optional	Present when the agent that answered the call is a Cisco BroadWorks subscriber.
<code>answeringCallId</code>	Optional	Present when the call offer from the call center to the agent does not contain a SIP call leg.
<code>answeringNetworkCallId</code>	Optional	Present when the call offer from the call center to the agent contains a SIP call leg.

6.4.17 Route Point Holiday Policy Applied

The *Route Point Holiday Policy Applied* event is produced when a call is treated according to the Holiday policy.

6.4.17.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `RoutePointHolidayPolicyAppliedEvent`. This event contains the common parameters listed at the beginning of this section.

The following table provides the list of parameters used.

Parameter	Parameter Type	Required	Description
<code>callId</code>	<code>CallId</code>	Yes	The call ID of the call.
<code>extTrackingId</code>	<code>ExternalTrackingId</code>	Yes	The external tracking ID of the call.
<code>routePointName</code>	<code>String</code>	Yes	Indicates the Route Point name (DNIS Name) associated with the RP number the call was received on.
<code>routePointNumber</code>	<code>Address</code>	Yes	Indicates the Route Point number (DNIS) the call was received on.
<code>remoteParty</code>	<code>PartyInformation</code>	Yes	Information about the remote party.
<code>redirect</code>	<code>Redirect</code>	No	Present if the call was redirected to another destination based on the Route Point Holiday policy. This happens when the policy action is set to "transfer". For more information, see section 8 Type Definitions .
<code>redirections</code>	<code>RedirectionList</code>	No	Present if a prior redirection took place. This parameter lists all the redirections of the received call. These are the details of other parties that have redirected this call prior to reaching the route point.
<code>treatmentStarted</code>	<code>EmptyContent</code>	No	Present if the policy starts playing an announcement or ringback according to the policy configuration.

6.4.18 Route Point Holiday Policy Treatment Completed

The *Route Point Holiday Policy Treatment Completed* event is produced when the announcement is completed for a call treated according to the Holiday policy.

6.4.18.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `RoutePointHolidayPolicyTreatmentCompletedEvent`. This event contains the common parameters listed at the beginning of this section.

The following table provides the list of parameters used.

Parameter	Parameter Type	Required	Description
<code>callId</code>	<code>CallId</code>	Yes	The call ID of the call.
<code>extTrackingId</code>	<code>ExternalTrackingId</code>	Yes	The external tracking ID of the call.

Parameter	Parameter Type	Required	Description
routePointName	String	Yes	Indicates the Route Point name (DNIS Name) associated with the Route Point number the call was received on.
routePointNumber	Address	Yes	Indicates the Route Point number (DNIS) the call was received on.
remoteParty	PartyInformation	Yes	Information about the remote party.
<i>redirect</i>	Redirect	No	Present if the call was redirected to another destination based on the Route Point Holiday policy. This happens when the policy action is set to "transfer". For more information, see section 8 Type Definitions .

6.4.19 Route Point Night Policy Applied

The *Route Point Night Policy Applied* event is produced when a call is treated according to the Night policy.

6.4.19.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `RoutePointNightPolicyAppliedEvent`. This event contains the common parameters listed at the beginning of this section.

The following table provides the list of parameters used.

Parameter	Parameter Type	Required	Description
<i>callId</i>	CallId	Yes	The call ID of the call.
<i>extTrackingId</i>	ExternalTrackingId	Yes	The external tracking ID of the call.
<i>routePointName</i>	String	Yes	Indicates the Route Point name (DNIS Name) associated with the RP number the call was received on.
<i>routePointNumber</i>	Address	Yes	Indicates the Route Point number (DNIS) the call was received on.
<i>remoteParty</i>	PartyInformation	Yes	Information about the remote party.
<i>redirect</i>	Redirect	No	Present if the call was redirected to another destination based on the Night policy. This happens when the policy action is set to "transfer". For more information, see section 8 Type Definitions .
<i>redirections</i>	RedirectionList	No	Present if a prior redirection took place. This parameter lists all the redirections of the received call. These are the details of other parties that have redirected this call prior to reaching the route point.
<i>treatmentStarted</i>	EmptyContent	No	Present if the policy starts playing an announcement or ringback according to the policy configuration.

6.4.20 Route Point Night Policy Treatment Completed

The *Route Point Night Policy Treatment Completed* event is produced when a call is treated according to the Night policy.

6.4.20.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `RoutePointNightPolicyTreatmentCompletedEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>callId</code>	<code>CallId</code>	Yes	The call ID of the call.
<code>extTrackingId</code>	<code>ExternalTrackingId</code>	Yes	The external tracking ID of the call.
<code>routePointName</code>	<code>String</code>	Yes	Indicates the Route Point name (DNIS Name) associated with the RP number the call was received on.
<code>routePointNumber</code>	<code>Address</code>	Yes	Indicates the Route Point number (DNIS) the call was received on.
<code>remoteParty</code>	<code>PartyInformation</code>	Yes	Information about the remote party.
<code>redirect</code>	<code>Redirect</code>	No	Present if the call was redirected to another destination based on the Night policy. This happens when the policy action is set to "transfer". For more information, see section 8 Type Definitions .
<code>reason</code>	<code>TreatmentCompletionReason</code>	Yes	

6.4.21 Route Point Treatment Started

The *Route Point Treatment Started* event is produced when a Route Point Play Treatment is started. The event specifies the `playCollectId` of the operation.

6.4.21.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `RoutePointTreatmentStartedEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>queueEntry</code>	<code>RoutePointQueueEntry</code>	Yes	The details of the <code>RoutePointQueueEntry</code> . A <code>RoutePointQueueEntry</code> contains information about a call in the Route Point queue. For more information, see section 8 Type Definitions .

The following table provides the list of parameters used in the `queueEntry` by this event.

Parameter	Mandatory or Optional	Comments
<code>callId</code>	Mandatory	
<code>extTrackingId</code>	Mandatory	
<code>remoteParty</code>	Mandatory	

Parameter	Mandatory or Optional	Comments
<i>addTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>playCollectInfo</i>	Mandatory	
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

6.4.22 Route Point Treatment Completed

The *Route Point Treatment Completed* event is produced when a Route Point Play Treatment is completed. The PlayCollectResult parameters contain the detail of the playback and DTMF data collection (if any) while the RoutePointQueueEntry identifies the call in the queue.

6.4.22.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointTreatmentCompletedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The details of the RoutePointQueueEntry. A RoutePointQueueEntry contains information about a call in the Route Point queue. For more information, see section 8 Type Definitions .
<i>playCollectResult</i>	PlayCollectResult	Yes	The details of the play and DTMF collection request performed. For more information, see section 8 Type Definitions .

The following table provides the list of parameters used in the queueEntry by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	

Parameter	Mandatory or Optional	Comments
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

The following table provides the list of parameters used in the playCollectResult by this event.

Parameter	Mandatory or Optional	Comments
<i>playCollectId</i>	Mandatory	
<i>reason</i>	Mandatory	
<i>digits</i>	Optional	Present if a DTMF input was collected.
<i>errorReason</i>	Optional	Present when the reason is set to "Error".

6.4.23 Route Point MOH Started

The *Route Point MOH Started* event is produced when a Route Point MOH Treatment is started. The event specifies the playCollectId of the operation.

6.4.23.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointMOHStartedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The details of the RoutePointQueueEntry. A RoutePointQueueEntry contains information about a call in the Route Point queue. For more information, see section 8 Type Definitions .

The following table provides the list of parameters used in the queueEntry by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>playCollectInfo</i>	Mandatory	
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.

Parameter	Mandatory or Optional	Comments
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

6.4.24 Route Point MOH Completed

The *Route Point MOH Completed* event is produced when a Route Point Play MOH is completed.

6.4.24.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `RoutePointMOHCompletedEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	<code>RoutePointQueueEntry</code>	Yes	The details of the <code>RoutePointQueueEntry</code> . A <code>RoutePointQueueEntry</code> contains information about a call in the Route Point queue. For more information, see section 8 Type Definitions .
<i>playCollectResult</i>	<code>PlayCollectResult</code>	Yes	The details of the play and DTMF collection request performed. For more information, see section 8 Type Definitions .

The following table provides the list of parameters used in the `queueEntry` by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

The following table provides the list of parameters used in the `playCollectResult` by this event.

Parameter	Mandatory or Optional	Comments
<i>playCollectId</i>	Mandatory	
<i>reason</i>	Mandatory	
<i>digits</i>	Optional	Present if a DTMF input was collected.
<i>errorReason</i>	Optional	Present when the reason is set to "Error"

6.4.25 Route Point Ringback Started

The *Route Point Ringback Started* event is produced when a Route Point MOH Treatment is started. The event specifies the playCollectId of the operation.

6.4.25.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointRingbackStartedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The details of the RoutePointQueueEntry. A RoutePointQueueEntry contains information about a call in the Route Point queue. For more information, see section 8 Type Definitions .

The following table provides the list of parameters used in the queueEntry by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>playCollectInfo</i>	Mandatory	
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

6.4.26 Route Point Ringback Completed

The *Route Point Ringback Completed* event is produced when a Route Point Play Ringback is completed. The PlayCollectResult parameters contain the detail of the playback and DTMF data collection (if any) while the RoutePointQueueEntry identifies the call in the queue.

6.4.26.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointRingbackCompletedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The details of the RoutePointQueueEntry. A RoutePointQueueEntry contains information about a call in the Route Point queue. For more information, see section 8 Type Definitions .
<i>playCollectResult</i>	PlayCollectResult	Yes	The details of the play and DTMF collection request performed. For more information, see section 8 Type Definitions .

The following table provides the list of parameters used in the queueEntry by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

The following table provides the list of parameters used in the playCollectResult by this event.

Parameter	Mandatory or Optional	Comments
<i>playCollectId</i>	Mandatory	
<i>reason</i>	Mandatory	
<i>digits</i>	Optional	Present if a DTMF input was collected.
<i>errorReason</i>	Optional	Present when the reason is set to "Error".



6.4.27 Route Point Busy Started

The *Route Point Busy Started* event is produced when a Route Point Busy Treatment is started. The event specifies the playCollectId of the operation.

6.4.27.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointBusyStartedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The details of the RoutePointQueueEntry. A RoutePointQueueEntry contains information about a call in the Route Point queue. For more information, see section 8 Type Definitions .

The following table provides the list of parameters used in the queueEntry by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>playCollectInfo</i>	Mandatory	
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

6.4.28 Route Point Busy Completed

The *Route Point Busy Completed* event is produced when a Route Point Play Busy is completed.

6.4.28.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointBusyCompletedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The details of the RoutePointQueueEntry. A RoutePointQueueEntry contains information about a call in the Route Point queue. For more information, see section 8 Type Definitions .
<i>playCollectResult</i>	PlayCollectResult	Yes	The details of the play and DTMF collection request performed. For more information, see section 8 Type Definitions .

The following table provides the list of parameters used in the queueEntry by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

The following table provides the list of parameters used in the playCollectResult by this event.

Parameter	Mandatory or Optional	Comments
<i>playCollectId</i>	Mandatory	
<i>reason</i>	Mandatory	
<i>digits</i>	Optional	Present if a DTMF input was collected.
<i>errorReason</i>	Optional	Present when the reason is set to "Error".

6.4.29 Route Point Silence Started

The *Route Point Silence Started* event is produced when a Route Point Play Silence is started. The event specifies the playCollectId of the operation.

6.4.29.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointSilenceStartedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The details of the RoutePointQueueEntry. A RoutePointQueueEntry contains information about a call in the Route Point queue. For more information, see section 8 Type Definitions .

The following table provides the list of parameters used in the queueEntry by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>playCollectInfo</i>	Mandatory	
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

6.4.30 Route Point Silence Completed

The *Route Point Silence Completed* event is produced when a Route Point Play Silence is completed.

6.4.30.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointSilenceCompletedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The details of the RoutePointQueueEntry. A RoutePointQueueEntry contains information about a call in the Route Point queue. For more information, see section 8 Type Definitions .

Parameter	Parameter Type	Required	Description
<i>playCollectResult</i>	PlayCollectResult	Yes	The details of the play and DTMF collection request performed. For more information, see section 8 Type Definitions .

The following table provides the list of parameters used in the queueEntry by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.

The following table provides the list of parameters used in the playCollectResult by this event.

Parameter	Mandatory or Optional	Comments
<i>playCollectId</i>	Mandatory	
<i>reason</i>	Mandatory	
<i>digits</i>	Optional	Present if a DTMF input was collected.
<i>errorReason</i>	Optional	Present when the reason is set to "Error".

6.4.31 Route Point Whisper Started

The *Route Point Whisper Started* event is produced when a whisper message is started.

6.4.31.1 Event Type Definition

Cisco BroadWorks issues this event by sending a RoutePointWhisperStartedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	RoutePointQueueEntry	Yes	The details of the RoutePointQueueEntry. A RoutePointQueueEntry contains information about a call in the Route Point queue. For more information, see section 8 Type Definitions .



The following table provides the list of parameters used in the queueEntry by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>routePointNumber</i>	Mandatory	
<i>routePointName</i>	Mandatory	
<i>outgoingCall</i>	Optional	Present when the call was added as a result of an outgoing Dial request.
<i>outgoingCallAnswerTime</i>	Optional	Present for outgoing calls that have been answered by the called destination.
<i>answeringUserId</i>	Optional	Present when the agent that answered the call is a Cisco BroadWorks subscriber.
<i>answeringCallId</i>	Optional	Present when the call offer from the call center to the agent does not contain a SIP call leg.
<i>answeringNetworkCallId</i>	Optional	Present when the call offer from the call center to the agent contains a SIP call leg.

6.4.32 Route Point Failed

The *Route Point Failed* event is produced when a Route Point enters the *Failed* state.

6.4.32.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *RoutePointFailedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>stateChangeReason</i>	StateChangeReason	Yes	The reason the Route Point entered the <i>Failed</i> state. For more information, see section 8 Type Definitions .

6.4.33 Route Point Recovered

The *Route Point Recovered* event is produced when a Route Point enters the *Normal* state.

6.4.33.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *RoutePointRecoveredEvent*. This event contains no parameter besides the common parameters listed at the beginning of this section.

6.5 ACD Events (**AS only**)

An ACD event is an event that reports a change associated to a call maintained in an ACD queue. The next subsections provide detailed information for all the available ACD event types:

- ACD Call Added
- ACD Call Offered to Agent
- ACD Call Answered by Agent
- ACD Call Abandoned
- ACD Call Released
- ACD Call Overflowed
- ACD Call Overflowed Treatment Completed
- ACD Call Transferred
- ACD Call Updated
- ACD Call Bounced
- ACD Call Forwarded
- ACD Call Forwarded Treatment Completed
- ACD Call Reordered
- ACD Call Promoted
- ACD Call Stranded
- ACD Call Stranded Treatment Completed
- ACD Call Escaped
- ACD Subscription
- ACD Subscription Resync
- ACD Holiday Policy Applied
- ACD Holiday Policy Treatment Completed
- ACD Night Policy Applied
- ACD Night Policy Treatment Completed
- ACD Whisper Started

6.5.1 ACD Call Added

The *ACD Call Added* event is produced when a new call is added to the ACD queue. For information on incoming call processing for details of what makes a call eligible to be added to the queue, see section [4.4.2 Incoming Call Processing](#).

6.5.1.1 Event Type Definition

Cisco BroadWorks issues this event by sending an ACDCallAddedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	ACDQueueEntry	Yes	The details of the ACDQueueEntry. An ACDQueueEntry contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .
<i>position</i>	PositiveInteger	Yes	The location in the queue where the call was added.
<i>redirections</i>	RedirectionList	No	Present if a prior redirection took place. This parameter lists all the redirections of the received call. These are the details of other parties that have redirected this call prior to reaching the ACD.

The following table provides the list of parameters used in the queueEntry by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>mandatoryEntrance</i>	Optional	Present when the call is being played the mandatory entrance message.
<i>preservedWaitTime</i>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<i>ACDNumber</i>	Mandatory	
<i>ACDName</i>	Mandatory	
<i>ACDPriority</i>	Optional	Set when the ACD is of type Premium.
<i>addTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium.
<i>preservedWaitTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium. It is only present when different from zero.

6.5.2 ACD Call Offered to Agent

The *ACD Call Offered* event is produced when the ACD distributes the call to an agent. This event indicates that the call is offered to the agent but remains in the queue.

6.5.2.1 Event Type Definition

Cisco BroadWorks issues this event by sending an ACDCallOfferedToAgentEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	ACDQueueEntry	Yes	The details of the ACDQueueEntry. An ACDQueueEntry contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .

The following table provides the list of parameters used in the queueEntry by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>reordered</i>	Optional	Present when the call has been reordered within the queue.
<i>preservedWaitTime</i>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<i>ACDNumber</i>	Mandatory	
<i>ACDName</i>	Mandatory	
<i>ACDPriority</i>	Optional	Set when the ACD is of type Premium.
<i>addTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium.
<i>preservedWaitTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium. It is only present when different from zero.

6.5.3 ACD Call Answered by Agent

The *ACD Call Answered* event is produced after a call has been offered to an agent, the agent has decided to accept the call, and the call has been removed from the queue.

6.5.3.1 Event Type Definition

Cisco BroadWorks issues this event by sending an `ACDCallAnsweredByAgentEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>queueEntry</code>	<code>ACDQueueEntry</code>	Yes	The details of the <code>ACDQueueEntry</code> . An <code>ACDQueueEntry</code> contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .

The following table provides the list of parameters used in the `queueEntry` by this event.

Parameter	Mandatory or Optional	Comments
<code>callId</code>	Mandatory	
<code>extTrackingId</code>	Mandatory	
<code>remoteParty</code>	Mandatory	
<code>addTime</code>	Mandatory	
<code>removeTime</code>	Mandatory	
<code>bounced</code>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<code>reordered</code>	Optional	Present when the call has been reordered within the queue.
<code>preservedWaitTime</code>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<code>ACDNumber</code>	Mandatory	
<code>ACDName</code>	Mandatory	
<code>ACDPriority</code>	Optional	Set when the ACD is of type Premium.
<code>addTimeInPriorityBucket</code>	Optional	Set when the ACD is of type Premium.
<code>preservedWaitTimeInPriorityBucket</code>	Optional	Set when the ACD is of type Premium. It is only present when different from zero.
<code>answeringUserId</code>	Optional	Present when the agent that answered the call is a Cisco BroadWorks subscriber.
<code>answeringCallId</code>	Optional	Present when the call offer from the call center to the agent does not contain a SIP call leg.

Parameter	Mandatory or Optional	Comments
<i>answeringNetworkCallId</i>	Optional	Present when the call offer from the call center to the agent contains a SIP call leg.

6.5.4 ACD Whisper Started

The *ACD Whisper Started* event is produced when a whisper message is started

6.5.4.1 Event Type Definition

Cisco BroadWorks issues this event by sending an *ACDWhisperStartedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	<i>ACDQueueEntry</i>	Yes	The details of the <i>ACDQueueEntry</i> . An <i>ACDQueueEntry</i> contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .

The following table provides the list of parameters used in the *queueEntry* by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>reordered</i>	Optional	Present when the call has been reordered within the queue.
<i>preservedWaitTime</i>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<i>ACDNumber</i>	Mandatory	
<i>ACDName</i>	Mandatory	
<i>ACDPriority</i>	Optional	Set when the ACD is of type Premium.
<i>addTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium.
<i>preservedWaitTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium. It is only present when different from zero.
<i>answeringUserId</i>	Optional	Present when the agent answered the call is a Cisco BroadWorks subscriber.

Parameter	Mandatory or Optional	Comments
<i>answeringCallId</i>	Optional	Present when the call offer from the call center to the agent does not contain a SIP call leg.
<i>answeringNetworkCallId</i>	Optional	Present when the call offer from the call center to the agent contains a SIP call leg.

6.5.5 ACD Call Overflowed

The *ACD Call Overflowed* event is produced after a call is marked as overflowed. The *overFlowReason* specifies the cause of the overflow.

6.5.5.1 Event Type Definition

Cisco BroadWorks issues this event by sending an *ACDCallOverflowedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	<i>ACDQueueEntry</i>	Yes	The details of the <i>ACDQueueEntry</i> . An <i>ACDQueueEntry</i> contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .
<i>overFlowReason</i>	<i>OverFlowReason</i>	Yes	Specify the reason for overflow. For more information, see section 8 Type Definitions .
<i>redirect</i>	<i>Redirect</i>	No	Present if the call was redirected to another destination based on the call overflowed policy. This happens when the policy action is set to "transfer". For more information, see section 8 Type Definitions .
<i>redirections</i>	<i>RedirectionList</i>	No	Present if a prior redirection took place. This parameter lists all the redirections of the received call. These are the details of other parties that have redirected this call prior to reaching the ACD. The parameter is not present when the call is overflowed because it stayed in the queue for too long (time overflow). This is because this event is issued when a call is already inside the queue and redirection information has already been provided by the <i>ACD Call Added</i> event.
<i>treatmentStarted</i>	<i>EmptyContent</i>	No	Present if the policy starts playing an announcement or ringback according to the policy configuration.

The following table provides the list of parameters used in the *queueEntry* by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	

Parameter	Mandatory or Optional	Comments
<i>removeTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>reordered</i>	Optional	Present when the call has been reordered within the queue.
<i>preservedWaitTime</i>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<i>ACDNumber</i>	Mandatory	
<i>ACDName</i>	Mandatory	
<i>ACDPriority</i>	Optional	Set when the ACD is of type Premium.
<i>addTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium.
<i>preservedWaitTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium. It is only present when different from zero.

6.5.6 ACD Call Overflowed Treatment Completed

The *ACD Call Overflowed Treatment Completed* event is produced when the announcement is completed for a call treated according to the Overflow policy. The *overFlowReason* specifies the cause of the overflow.

6.5.6.1 Event Type Definition

Cisco BroadWorks issues this event by sending an *ACDCallOverflowedTreatmentCompletedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	ACDQueueEntry	Yes	The details of the ACDQueueEntry. An ACDQueueEntry contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .
<i>overFlowReason</i>	OverFlowReason	Yes	Specify the reason for overflow. For more information, see section 8 Type Definitions .
<i>redirect</i>	Redirect	No	Present if the call was redirected to another destination based on the call overflowed policy. This happens when the policy action is set to "transfer". For more information, see section 8 Type Definitions .



Parameter	Parameter Type	Required	Description
<i>reason</i>	TreatmentCompletionReason	Yes	

The following table provides the list of parameters used in the *queueEntry* by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>removeTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>reordered</i>	Optional	Present when the call has been reordered within the queue.
<i>preservedWaitTime</i>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<i>ACDNumber</i>	Mandatory	
<i>ACDName</i>	Mandatory	
<i>ACDPriority</i>	Optional	Set when the ACD is of type Premium.
<i>addTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium.
<i>preservedWaitTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium. It is only present when different from zero.

6.5.7 ACD Call Abandoned

The *ACD Call Abandoned* event is produced after a call has been abandoned by a remote application (for example, Cisco BroadWorks client).

6.5.7.1 Event Type Definition

Cisco BroadWorks issues this event by sending an ACDCallAbandonedEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	ACDQueueEntry	Yes	The details of the ACDQueueEntry. An ACDQueueEntry contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .
<i>entrancePlaying</i>	EmptyContent	No	Indicates, when present, that the entrance message was playing at the time the call was abandoned.

The following table provides the list of parameters used in the queueEntry by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>removeTime</i>	Mandatory	
<i>mandatoryEntrance</i>	Optional	Present when the call is being played the mandatory entrance message.
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>reordered</i>	Optional	Present when the call has been reordered within the queue.
<i>preservedWaitTime</i>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<i>ACDNumber</i>	Mandatory	
<i>ACDName</i>	Mandatory	
<i>ACDPriority</i>	Optional	Set when the ACD is of type Premium.
<i>addTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium.
<i>preservedWaitTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium. It is only present when different from zero.

6.5.8 ACD Call Transferred

The *ACD Call Transferred* event is produced when the call was removed from the queue as it is transferred to a new destination following an ACD Blind Transfer.

6.5.8.1 Event Type Definition

Cisco BroadWorks issues this event by sending an ACDCallTransferredEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	ACDQueueEntry	Yes	The details of the ACDQueueEntry. An ACDQueueEntry contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .
<i>redirect</i>	Redirect	Yes	The detail of the redirect. For more information, see section 8 Type Definitions .

The following table provides the list of parameters used in the queueEntry by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>removeTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>reordered</i>	Optional	Present when the call has been reordered within the queue.
<i>preservedWaitTime</i>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<i>ACDNumber</i>	Mandatory	
<i>ACDName</i>	Mandatory	
<i>ACDPriority</i>	Optional	Set when the ACD is of type Premium.
<i>addTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium.
<i>preservedWaitTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium. It is only present when different from zero.

6.5.9 ACD Call Updated

The *ACD Call Updated* event is generated in any of the following cases:

- When the external tracking id is changed, this can happen for instance following a consultation transfer.
- When the remote party information are being modified.

6.5.9.1 Event Type Definition

Cisco BroadWorks issues this event by sending an `ACDCallUpdatedEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>queueEntry</code>	<code>ACDQueueEntry</code>	Yes	The details of the <code>ACDQueueEntry</code> . An <code>ACDQueueEntry</code> contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .
<code>reason</code>	<code>CallUpdateReason</code>	Yes	Specify the reason for the update.

The following table provides the list of parameters used in the `queueEntry` by this event.

Parameter	Mandatory or Optional	Comments
<code>callId</code>	Mandatory	
<code>extTrackingId</code>	Mandatory	
<code>remoteParty</code>	Mandatory	
<code>addTime</code>	Mandatory	
<code>mandatoryEntrance</code>	Optional	Present when the call is being played the mandatory entrance message.
<code>bounced</code>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<code>reordered</code>	Optional	Present when the call has been reordered within the queue.
<code>preservedWaitTime</code>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<code>ACDNumber</code>	Mandatory	
<code>ACDName</code>	Mandatory	
<code>ACDPriority</code>	Optional	Set when the ACD is of type Premium.

Parameter	Mandatory or Optional	Comments
<i>addTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium.
<i>preservedWaitTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium. It is only present when different from zero.

6.5.10 ACD Call Bounced

The *ACD Call Bounced* event is produced after a call is bounced. When a call is bounced, the bounced parameter is added to the ACDQueueEntry reported in the event. This parameter will be maintained in all future events until the call is finally removed from the ACD queue.

6.5.10.1 Event Type Definition

Cisco BroadWorks issues this event by sending an *ACDCallBouncedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	ACDQueueEntry	Yes	The details of the ACDQueueEntry. An ACDQueueEntry contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .
<i>position</i>	Long	Yes	The location in the queue where the call was added.
<i>redirect</i>	Redirect	No	Present if the call was redirected to another destination based on the bounced call policy. This happens when the policy action is set to "transfer". For more information, see section 8 Type Definitions .

The following table provides the list of parameters used in the *queueEntry* by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Mandatory	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>reordered</i>	Optional	Present when the call has been reordered within the queue.
<i>preservedWaitTime</i>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<i>ACDNumber</i>	Mandatory	



Parameter	Mandatory or Optional	Comments
<i>ACDName</i>	Mandatory	
<i>ACDPriority</i>	Optional	Set when the ACD is of type Premium.
<i>addTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium.
<i>preservedWaitTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium.

6.5.11 ACD Call Forwarded

The *ACD Call Forwarded* event is produced when a call is forwarded following the application of the Forced Forwarding policy.

6.5.11.1 Event Type Definition

Cisco BroadWorks issues this event by sending an *ACDCallForwardedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>callId</i>	CallId	Yes	The call ID of the call.
<i>extTrackingId</i>	ExternalTrackingId	Yes	The external tracking ID of the call.
<i>remoteParty</i>	PartyInformation	Yes	Information about the remote party.
<i>acdName</i>	String	Yes	Indicates the ACD name (DNIS Name) associated with the <i>acdNumber</i> the call was received on.
<i>acdNumber</i>	Address	Yes	Indicates the ACD number (DNIS) the call was received on.
<i>redirect</i>	Redirect	Yes	Present if the call was redirected to another destination based on the call forwarded policy. For more information, see section 8 Type Definitions .
<i>redirections</i>	RedirectionList	No	Present if a prior redirection took place. This parameter lists all the redirections of the received call. These are the details of other parties that have redirected this call prior to reaching the ACD.
<i>treatmentStarted</i>	EmptyContent	No	Present if the policy starts playing an announcement or ringback according to the policy configuration.



6.5.12 ACD Call Forwarded Treatment Completed

The *ACD Call Forwarded Treatment Completed* event is produced when the announcement is completed for a call treated according to the Forced Forwarding policy.

6.5.12.1 Event Type Definition

Cisco BroadWorks issues this event by sending an `ACDCallForwardedTreatmentCompletedEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>callId</code>	<code>CallId</code>	Yes	The call ID of the call.
<code>extTrackingId</code>	<code>ExternalTrackingId</code>	Yes	The external tracking ID of the call.
<code>remoteParty</code>	<code>PartyInformation</code>	Yes	Information about the remote party.
<code>acdName</code>	<code>String</code>	Yes	Indicates the ACD name (DNIS Name) associated with the <code>AcdNumber</code> the call was received on.
<code>acdNumber</code>	<code>Address</code>	Yes	Indicates the ACD number (DNIS) the call was received on.
<code>redirect</code>	<code>Redirect</code>	Yes	Present if the call was redirected to another destination based on the call forwarded policy. For more information, see section 8 Type Definitions .
<code>reason</code>	<code>TreatmentCompletionReason</code>	Yes	

6.5.13 ACD Call Released

The *ACD Call Released* event is generated following an ACD Call Released request.

6.5.13.1 Event Type Definition

Cisco BroadWorks issues this event by sending an `ACDCallReleasedEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>queueEntry</code>	<code>ACDQueueEntry</code>	Yes	The details of the <code>ACDQueueEntry</code> . An <code>ACDQueueEntry</code> contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .

The following table provides the list of parameters used in the `queueEntry` by this event.

Parameter	Mandatory or Optional	Comments
<code>callId</code>	Mandatory	
<code>extTrackingId</code>	Mandatory	
<code>remoteParty</code>	Mandatory	
<code>addTime</code>	Mandatory	

Parameter	Mandatory or Optional	Comments
<i>removeTime</i>	Mandatory	
<i>mandatoryEntrance</i>	Optional	Present when the call is being played the mandatory entrance message.
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>reordered</i>	Optional	Present when the call has been reordered within the queue.
<i>preservedWaitTime</i>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<i>ACDNumber</i>	Mandatory	
<i>ACDName</i>	Mandatory	
<i>ACDPriority</i>	Optional	Set when the ACD is of type Premium.
<i>addTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium.
<i>preservedWaitTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium. It is only present when different from zero.

6.5.14 ACD Call Reordered

The *ACD Call Reordered* event is produced when a call is reordered (for example, through an ACD Call Reorder).

6.5.14.1 Event Type Definition

Cisco BroadWorks issues this event by sending an *ACDCallReorderedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	ACDQueueEntry	Yes	The details of the <i>ACDQueueEntry</i> . An <i>ACDQueueEntry</i> contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .
<i>position</i>	PositiveInteger	Yes	The new location of the call in the queue.

The following table provides the list of parameters used in the *queueEntry* by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	

Parameter	Mandatory or Optional	Comments
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>reordered</i>	Mandatory	Present when the call has been reordered within the queue.
<i>preservedWaitTime</i>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<i>ACDNumber</i>	Mandatory	
<i>ACDName</i>	Mandatory	
<i>ACDPriority</i>	Optional	Set when the ACD is of type Premium.
<i>addTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium.
<i>preservedWaitTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium. It is only present when different from zero.

6.5.15 ACD Call Promoted

The *ACD Call Promoted* event is produced when a call has been promoted to another priority either manually (for example, through an ACD Promote Call) or automatically. The *ACDPriority* parameter is set to the new priority. The *position* parameter is set to the new location of the call in the queue.

6.5.15.1 Event Type Definition

Cisco BroadWorks issues this event by sending an *ACDCallPromotedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	<i>ACDQueueEntry</i>	Yes	The details of the <i>ACDQueueEntry</i> . An <i>ACDQueueEntry</i> contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .
<i>position</i>	<i>PositiveInteger</i>	Yes	The new location of the call in the queue.

The following table provides the list of parameters used in the *queueEntry* by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	

Parameter	Mandatory or Optional	Comments
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>reordered</i>	Mandatory	Present when the call has been reordered within the queue.
<i>preservedWaitTime</i>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<i>ACDNumber</i>	Mandatory	
<i>ACDName</i>	Mandatory	
<i>ACDPriority</i>	Mandatory	Set when the ACD is of type Premium.
<i>addTimeInPriorityBucket</i>	Mandatory	Set when the ACD is of type Premium.
<i>preservedWaitTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium. It is only present when different from zero.

6.5.16 ACD Call Stranded

The *ACD Call Stranded* event is produced when a call is identified as stranded as explained in section [4.4.2.2 Stranded Call Policy](#). The call is removed from the queue and handled as specified by the Stranded Call policy.

6.5.16.1 Event Type Definition

Cisco BroadWorks issues this event by sending an `ACDCallStrandedEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	<code>ACDQueueEntry</code>	Yes	The details of the <code>ACDQueueEntry</code> . An <code>ACDQueueEntry</code> contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .
<i>redirect</i>	<code>Redirect</code>	No	Present if the call was redirected to another destination based on the stranded call policy. This happens when the policy action is set to "transfer". For more information, see section 8 Type Definitions .
<i>treatmentStarted</i>	<code>EmptyContent</code>	No	Present if the policy starts playing an announcement or ringback according to the policy configuration.

The following table provides the list of parameters used in the queueEntry by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>removeTime</i>	Mandatory	
<i>mandatoryEntrance</i>	Optional	Present when the call is being played the mandatory entrance message.
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>reordered</i>	Optional	Present when the call has been reordered within the queue.
<i>preservedWaitTime</i>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<i>ACDNumber</i>	Mandatory	
<i>ACDName</i>	Mandatory	
<i>ACDPriority</i>	Optional	Set when the ACD is of type Premium.
<i>addTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium.
<i>preservedWaitTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium. It is only present when different from zero.

6.5.17 ACD Call Stranded Treatment Completed

The *ACD Call Stranded Treatment Completed* event is produced when the announcement is completed for a call treated according to the Stranded Call policy.

6.5.17.1 Event Type Definition

Cisco BroadWorks issues this event by sending an *ACDCallStrandedTreatmentCompletedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	ACDQueueEntry	Yes	The details of the ACDQueueEntry. An ACDQueueEntry contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .



Parameter	Parameter Type	Required	Description
<i>redirect</i>	Redirect	No	Present if the call was redirected to another destination based on the stranded call policy. This happens when the policy action is set to "transfer". For more information, see section 8 Type Definitions .
<i>reason</i>	TreatmentCompletionReason	Yes	

The following table provides the list of parameters used in the queueEntry by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>removeTime</i>	Mandatory	
<i>mandatoryEntrance</i>	Optional	Present when the call is being played the mandatory entrance message.
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>reordered</i>	Optional	Present when the call has been reordered within the queue.
<i>preservedWaitTime</i>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<i>ACDNumber</i>	Mandatory	
<i>ACDName</i>	Mandatory	
<i>ACDPriority</i>	Optional	Set when the ACD is of type Premium.
<i>addTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium.
<i>preservedWaitTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium. It is only present when different from zero.

6.5.18 ACD Call Stranded Unavailable

The *ACD Call Stranded Unavailable* event is produced when a call is identified as stranded unavailable as explained in section [4.4.2.3 Stranded Unavailable Call Policy](#). The call is removed from the queue and handled as specified by the Stranded Unavailable Call policy.

6.5.18.1 Event Type Definition

Cisco BroadWorks issues this event by sending an `ACDCallStrandedUnavailableEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>queueEntry</code>	<code>ACDQueueEntry</code>	Yes	The details of the <code>ACDQueueEntry</code> . An <code>ACDQueueEntry</code> contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .
<code>redirect</code>	<code>Redirect</code>	No	Present if the call was redirected to another destination based on the stranded unavailable call policy. This happens when the policy action is set to "transfer". For more information, see section 8 Type Definitions .
<code>treatmentStarted</code>	<code>EmptyContent</code>	No	Present if the policy starts playing an announcement or ringback according to the policy configuration.

The following table provides the list of parameters used in the `queueEntry` by this event.

Parameter	Mandatory or Optional	Comments
<code>callId</code>	Mandatory	
<code>extTrackingId</code>	Mandatory	
<code>remoteParty</code>	Mandatory	
<code>addTime</code>	Mandatory	
<code>removeTime</code>	Mandatory	
<code>mandatoryEntrance</code>	Optional	Present when the call is being played the mandatory entrance message.
<code>bounced</code>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<code>reordered</code>	Optional	Present when the call has been reordered within the queue.
<code>preservedWaitTime</code>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<code>ACDNumber</code>	Mandatory	
<code>ACDName</code>	Mandatory	
<code>ACDPriority</code>	Optional	Set when the ACD is of type <i>Premium</i> .

Parameter	Mandatory or Optional	Comments
<i>addTimeInPriorityBucket</i>	Optional	Set when the ACD is of type <i>Premium</i> .
<i>preservedWaitTimeInPriorityBucket</i>	Optional	Set when the ACD is of type <i>Premium</i> . It is only present when different from zero.

6.5.19 ACD Call Stranded Unavailable Treatment Completed

The *ACD Call Stranded Unavailable Treatment Completed* event is produced when the announcement is completed for a call treated according to the Stranded Unavailable Call policy.

6.5.19.1 Event Type Definition

Cisco BroadWorks issues this event by sending an `ACDCALLSTRANDEDUNAVAILABLETREATMENTCOMPLETEDEVENT`. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	<code>ACDQueueEntry</code>	Yes	The details of the <code>ACDQueueEntry</code> . An <code>ACDQueueEntry</code> contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .
<i>redirect</i>	<code>Redirect</code>	No	Present if the call was redirected to another destination based on the stranded unavailable call policy. This happens when the policy action is set to "transfer". For more information, see section 8 Type Definitions .
<i>reason</i>	<code>TreatmentCompletionReason</code>	Yes	

The following table provides the list of parameters used in the `queueEntry` by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>removeTime</i>	Mandatory	
<i>mandatoryEntrance</i>	Optional	Present when the call is being played the mandatory entrance message.
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>reordered</i>	Optional	Present when the call has been reordered within the queue.

Parameter	Mandatory or Optional	Comments
<i>preservedWaitTime</i>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<i>ACDNumber</i>	Mandatory	
<i>ACDName</i>	Mandatory	
<i>ACDPriority</i>	Optional	Set when the ACD is of type <i>Premium</i> .
<i>addTimeInPriorityBucket</i>	Optional	Set when the ACD is of type <i>Premium</i> .
<i>preservedWaitTimeInPriorityBucket</i>	Optional	Set when the ACD is of type <i>Premium</i> . It is only present when different from zero.

6.5.20 ACD Call Escaped

The *ACD Call Escaped* event is produced when the caller dial the configured escape digits while being in the queue. The call is removed from the queue and the caller is provided with Busy Treatment. If the queue is configured with the Call Forwarding Busy or the Voice Messaging service, then the call is handled accordingly.

6.5.20.1 Event Type Definition

Cisco BroadWorks issues this event by sending an *ACDCallEscapedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>queueEntry</i>	<i>ACDQueueEntry</i>	Yes	The details of the <i>ACDQueueEntry</i> . An <i>ACDQueueEntry</i> contains information about a call in the ACD queue. For more information, see section 8 Type Definitions .

The following table provides the list of parameters used in the *queueEntry* by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>extTrackingId</i>	Mandatory	
<i>remoteParty</i>	Mandatory	
<i>addTime</i>	Mandatory	
<i>removeTime</i>	Mandatory	
<i>mandatoryEntrance</i>	Optional	Present when the call is being played the mandatory entrance message.

Parameter	Mandatory or Optional	Comments
<i>bounced</i>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<i>reordered</i>	Optional	Present when the call has been reordered within the queue.
<i>preservedWaitTime</i>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<i>ACDNumber</i>	Mandatory	
<i>ACDName</i>	Mandatory	
<i>ACDPriority</i>	Optional	Set when the ACD is of type Premium.
<i>addTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium.
<i>preservedWaitTimeInPriorityBucket</i>	Optional	Set when the ACD is of type Premium. It is only present when different from zero.

6.5.21 ACD Subscription Event and ACD Subscription Resync Event

The *ACD Subscription* event is issued immediately after a subscription is added or refreshed, provided that the subscription target is not a collection of subscriber (for example, a group) and that contact was registered with the subscription.

The *ACD Subscription Resync* event is issued whenever the Application Server shuts down abnormally allowing a remote application to resynchronize.

Both the *ACD Subscription* event and the *ACD Subscription Resync* event share the same definition.

6.5.21.1 Event Type Definition

Cisco BroadWorks issues this event by sending an *ACDSUBSCRIPTIONEVENT* (or an *ACDSUBSCRIPTIONRESYNC*). This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>queueEntries</i>	ACDQueueEntryList	No	Contains the list of <i>ACDQueueEntry</i> maintained by the ACD. For more information, see section 8 Type Definitions . Only present if there is at least one <i>ACDQueueEntry</i> maintained by the ACD.

The order of the *queueEntry* in the list reflects the position in the queue. The following table provides the list of *queueEntry* parameters used by each *ACDQueueEntry* in the list.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	

Parameter	Mandatory or Optional	Comments
<code>extTrackingId</code>	Mandatory	
<code>remoteParty</code>	Mandatory	
<code>addTime</code>	Mandatory	
<code>mandatoryEntrance</code>	Optional	Present when the call is being played the mandatory entrance message.
<code>bounced</code>	Optional	Present when the call has been bounced once or more before. It marks calls that have been bounced so they can be easily identified.
<code>reordered</code>	Optional	Present when the call has been reordered within the queue.
<code>preservedWaitTime</code>	Optional	Present when the call has been waiting in another ACD queue and was transferred to this ACD queue. The preserved represents the total time spent waiting in the previous ACD queue(s).
<code>ACDNumber</code>	Mandatory	
<code>ACDName</code>	Mandatory	
<code>ACDPriority</code>	Optional	Set when the ACD is of type Premium.
<code>addTimeInPriorityBucket</code>	Optional	Set when the ACD is of type Premium.
<code>preservedWaitTimeInPriorityBucket</code>	Optional	Set when the ACD is of type Premium. It is only present when different from zero.

6.5.22 ACD Holiday Policy Applied

The *ACD Holiday Policy Applied* event is produced when a call is treated according to the Holiday policy.

6.5.22.1 Event Type Definition

Cisco BroadWorks issues this event by sending an `ACDHolidayPolicyAppliedEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>callId</code>	CallId	Yes	The call ID of the call.
<code>extTrackingId</code>	ExternalTrackingId	Yes	The external tracking ID of the call.
<code>remoteParty</code>	PartyInformation	Yes	Information about the remote party.
<code>acdName</code>	String	Yes	Indicates the ACD name (DNIS Name) associated with the <code>AcDNumber</code> the call was received on.
<code>acdNumber</code>	Address	Yes	Indicates the ACD number (DNIS) the call was received on.

Parameter	Parameter Type	Required	Description
<i>redirect</i>	Redirect	No	Present if the call was redirected to another destination based on the holiday policy. This happens when the policy action is set to “transfer”. For more information, see section 8 Type Definitions .
<i>redirections</i>	RedirectionList	No	Present if a prior redirection took place. This parameter lists all the redirections of the received call. These are the details of other parties that have redirected this call prior to reaching the ACD.
<i>treatmentStarted</i>	EmptyContent	No	Present if the policy starts playing an announcement or ringback according to the policy configuration.

6.5.23 ACD Holiday Policy Treatment Completed

The *ACD Holiday Policy Treatment Completed* event is produced when announcement is completed for a call treated according to the Holiday policy.

6.5.23.1 Event Type Definition

Cisco BroadWorks issues this event by sending an *ACDHolidayPolicyTreatmentCompletedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>callId</i>	CallId	Yes	The call ID of the call.
<i>extTrackingId</i>	ExternalTrackingId	Yes	The external tracking ID of the call.
<i>remoteParty</i>	PartyInformation	Yes	Information about the remote party.
<i>acdName</i>	String	Yes	Indicates the ACD name (DNIS Name) associated with the AcdNumber the call was received on.
<i>acdNumber</i>	Address	Yes	Indicates the ACD number (DNIS) the call was received on.
<i>redirect</i>	Redirect	No	Present if the call was redirected to another destination based on the holiday policy. This happens when the policy action is set to “transfer”. For more information, see section 8 Type Definitions .
<i>reason</i>	TreatmentCompletionReason	No	

6.5.24 ACD Night Policy Applied

The *ACD Night Policy Applied* event is produced when a call is treated according to the Night policy.

6.5.24.1 Event Type Definition

Cisco BroadWorks issues this event by sending an `ACDNightPolicyAppliedEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>callId</code>	CallId	Yes	The call ID of the call.
<code>extTrackingId</code>	ExternalTrackingId	Yes	The external tracking ID of the call.
<code>remoteParty</code>	PartyInformation	Yes	Information about the remote party.
<code>acdName</code>	String	Yes	Indicates the ACD name (DNIS Name) associated with the <code>AcdNumber</code> the call was received on.
<code>acdNumber</code>	Address	Yes	Indicates the ACD number (DNIS) the call was received on.
<code>redirect</code>	Redirect	No	Present if the call was redirected to another destination based on the night policy. This happens when the policy action is set to "transfer". For more information, see section 8 Type Definitions .
<code>redirections</code>	RedirectionList	No	Present if a prior redirection took place. This parameter lists all the redirections of the received call. These are the details of other parties that have redirected this call prior to reaching the ACD.
<code>treatmentStarted</code>	EmptyContent	No	Present if the policy starts playing an announcement or ringback according to the policy configuration.

6.5.25 ACD Night Policy Treatment Completed

The *ACD Night Policy Treatment Completed* event is produced when announcement is completed for a call treated according to the Night policy.

6.5.25.1 Event Type Definition

Cisco BroadWorks issues this event by sending an `ACDNightPolicyTreatmentCompletedEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>callId</code>	CallId	Yes	The call ID of the call.
<code>extTrackingId</code>	ExternalTrackingId	Yes	The external tracking ID of the call.
<code>remoteParty</code>	PartyInformation	Yes	Information about the remote party.

Parameter	Parameter Type	Required	Description
<i>acdName</i>	String	Yes	Indicates the ACD name (DNIS Name) associated with the <i>AcdNumber</i> the call was received on.
<i>acdNumber</i>	Address	Yes	Indicates the ACD number (DNIS) the call was received on.
<i>redirect</i>	Redirect	No	Present if the call was redirected to another destination based on the night policy. This happens when the policy action is set to "transfer". For more information, see section 8 Type Definitions .
<i>Reason</i>	TreatmentCompletionReason	No	

6.6 Agent Events (**AS only**)

An agent event is an event that reports a change associated to an agent. The next subsections provide detailed information for all the available agent event types:

- Agent State
- Agent Disposition Code Added
- ACD Agent Join Update

6.6.1 Agent State

The *Agent State* event is produced when an agent changes state or if there is a change in alert severity for a measurement pertaining to an agent. State change can be initiated by a remote application Agent State Change request, from the Cisco BroadWorks Call Center client, from the agent's endpoint, or from the web portal.

6.6.1.1 Event Type Definition

Cisco BroadWorks issues this event by sending an *AgentStateEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>agentStateInfo</i>	AgentStateInformation	Yes	Information on the agent state.

The following table provides the list of parameters used.

Parameter	Mandatory or Optional	Comments
<i>state</i>	Mandatory	
<i>stateTimestamp</i>	Mandatory	
<i>unavailableCode</i>	Optional	Present when the agent changes to the <i>Unavailable</i> state and when the unavailable codes are enabled.
<i>signInTimestamp</i>	Optional	Present when the agent is not in the <i>Sign-out</i> state. This is the time stamp value of the agent sign-in (that is, transition out the <i>Sign-out</i> state).

Parameter	Mandatory or Optional	Comments
<i>totalAvailableTime</i>	Optional	Present when the agent is not in the <i>Sign-out</i> state. This is the total available time since the last agent sign-in. Time currently spent in the <i>Available</i> state not included.
<i>averageWrapUpTime</i>	Optional	Present when the agent is not in the <i>Sign-out</i> state. This is the average wrap-up time per handled call since the last sign-in. Also indicates if the value is normal, and if not, the severity according to configured threshold.
<i>wrapUpCallId</i>	Optional	Present when the agent state changes to <i>Wrap-up</i> state. It provides the call ID of the last ACD call the agent was involved in.
<i>wrapUpCallCenterUserId</i>	Optional	Present when the agent is in <i>Wrap-up</i> state. It indicates the <i>subscriberId</i> of the ACD for the last ACD or outgoing ACD call made by the agent. This information element can be used to determine the appropriate list of disposition codes that can be entered by the agent in wrap-up.

6.6.2 Agent Disposition Code Added

The *Agent Disposition Code Added* event is produced when the agent enters a disposition code during the call or while in the *Wrap-up* state. A disposition code can be added using the Cisco BroadWorks Call Center client, the Agent's endpoint or via a remote application request.

If the ACD or Route Point is configured with a default disposition code, then the code is added to call events sent in the context of a call received from that ACD or Route Point, specifically within the *ACDCallInfo* or *ACDOutgoingCallInfo* parameter.

The *Agent Disposition Code Added* event is produced when an agent disposition code is explicitly added during the call or while in the *Wrap-up* state.

The *callId* parameter in the event identifies the call associated with the Disposition Code.

6.6.2.1 Event Type Definition

Cisco BroadWorks issues this event by sending an *AgentDispositionCodeAddedEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>dispositionCode</i>	AgentDispositionCode	Yes	The disposition code entered.
<i>callId</i>	CallId	Yes	This element is present when the agent state is in the <i>Wrap-up</i> state. It provides the call ID of the last ACD call the agent was involved in.

6.6.3 ACD Agent Join Update

The *ACD Agent Joining Update* event is produced when the agent joins or unjoins an ACD. An agent joins or unjoins an ACD using the Cisco BroadWorks Call Center client or the web portal.

6.6.3.1 Event Type Definition

Cisco BroadWorks issues this event by sending an `ACDAgentJoinUpdateEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>joinUpdateData</code>	<code>ACDAgentJoinUpdateData</code>	Yes	Specify if the agent is joining or not the ACD.

6.6.4 Agent Subscription

The *Agent Subscription* event is issued immediately after a subscription is added or refreshed, provided that the subscription target is not a collection of subscribers (for example, a group). For more information, see section [3.4.5 Event Subscription](#).

6.6.4.1 Event Type Definition

Cisco BroadWorks issues this event by sending an `AgentSubscriptionEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>joinData</code>	<code>ACDAgentJoinData</code>	Yes	Notification data for the ACD Agent Join.
<code>stateInfo</code>	<code>AgentStateInformation</code>	Yes	The agent's state. This is only present when agent state information is available.

6.7 Agent Monitoring Events (AS only)

Agent monitoring events are events that provide information related to the overall performance of an agent. The agent monitoring events are:

- Call Center Agent Monitoring
- Call Center Agent On Call Alert
- Call Center Agent Monitoring Subscription

6.7.1 Call Center Agent Monitoring

The *CallCenterAgentMonitoringEvent* allows a subscriber to receive notifications related to the overall performance of an agent. `CallCenterAgentMonitoringEvent` is generated when an agent releases or transfers an ACD call, at which time alert severity is evaluated.

This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>monitoringStatus</code>	<code>CallCenterAgentMonitoringStatus</code>	Yes	The current call center agent monitoring status.



The following table lists which parameters of the CallCenterMonitoringStatus are mandatory or optional.

Parameter	Mandatory or Optional	Comments
<i>averageACDCallTime</i>	Optional	Not present if the agent state is <i>Sign-out</i> . Also indicates if the value is normal, and if not, the severity according to configured threshold.
<i>averageACDOutgoingCallTime</i>	Optional	Not present if the agent state is <i>Sign-out</i> . Also indicates if the value is normal, and if not, the severity according to configured threshold.

6.7.2 Call Center Agent On Call Alert

The CallCenterAgentOnCallAlertEvent is sent to raise or clear alerts regarding an agent's on-call state. If the agent has been either idle or on a call for too long, according to configured thresholds, this event will indicate which alert condition prevails and with what severity.

This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>alertInfo</i>	CallCenterAgentOnCallAlertInfo	Yes	The current on-call/idle time and alert status.

The following table lists which parameters of the CallCenterAgentOnCallAlertInfo are mandatory or optional.

Parameter	Mandatory or Optional	Comments
<i>alertEvent</i>	Mandatory	Alert Type identifies whether the event pertains to In-call Time or Idle Time.
<i>Timestamp</i>	Mandatory	When alertEvent indicates "OnCallTime", contains the answer time of the longest call the agent is in. When alertEvent indicates "IdleTime", contains the release time of the call most recently released. If the agent receives a new call while Idle severity is greater than "Green", the "timestamp" is set to the new call's start time and the alert is cleared.

6.7.3 Call Center Agent Monitoring Subscription

An initial CallCenterAgentMonitoringSubscriptionEvent is sent upon subscription to the event package.

This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>monitoringStatus</i>	CallCenterAgentMonitoringStatus	Yes	The current call center agent monitoring status

Parameter	Parameter Type	Required	Description
<code>alertInfo</code>	CallCenterAgentOnCallAlertInfo	No	The current on-call/idle time and alert status. Omitted if the alerts are not enabled.

For more details on the parameters used by this event, see sections [6.7.1 Call Center Agent Monitoring](#) and [6.7.2 Call Center Agent On Call Alert](#).

6.8 ACD Configuration Events (AS only)

An ACD Configuration event is an event that reports a change associated to an ACD configuration. The next subsections provide detailed information for all the available ACD configuration event types:

- ACD Forced Forwarding
- ACD Holiday Service
- ACD Night Service
- ACD Configuration Subscription

6.8.1 ACD Forced Forwarding

The ACD Forced Forwarding event is generated when the ACD Forced Forwarding service configuration is modified.

6.8.1.1 Event Type Definition

Cisco BroadWorks issues this event by sending an ACDForcedForwardingEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>active</code>	Boolean	Yes	Indicate if the ACD Forced Forwarding configuration is active.

6.8.2 ACD Holiday Service

The ACD Holiday Service event is generated when the ACD Holiday service configuration is modified.

6.8.2.1 Event Type Definition

Cisco BroadWorks issues this event by sending an ACDHolidayServiceEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>active</code>	Boolean	Yes	Indicate if the ACD Holiday Service configuration is active. Activation occurs when the holiday period starts and deactivation occurs when the holiday period ends.

6.8.3 ACD Night Service

The ACD Night Service event is generated when the ACD Night service configuration is modified.

6.8.3.1 Event Type Definition

Cisco BroadWorks issues this event by sending an `ACDNightServiceEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>active</code>	Boolean	Yes	Indicate if the ACD Night Service configuration is active.

6.8.4 ACD Configuration Subscription

The ACD Configuration Subscription event is issued immediately after a subscription is added or refreshed, provided that the subscription target is not a collection of subscriber (for example, a group) and that contact was registered with the subscription.

6.8.4.1 Event Type Definition

Cisco BroadWorks issues this event by sending an `ACDConfigurationSubscriptionEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>forcedForwarding</code>	ActivatableServiceInfo	Yes	When present, indicates that ACD Forced Forwarding is active.
<code>holidayService</code>	ActivatableServiceInfo	Yes	When present, indicates that ACD Holiday Service is active.
<code>nightService</code>	ActivatableServiceInfo	Yes	When present, indicates that ACD Night Service is active.
<code>forceNightService</code>	EmptyContent	No	When present, indicates that the ACD Night Service was manually activated, overriding its normal schedule.

6.9 ACD Monitoring Events (AS only)

An ACD monitoring event is an event that provides information related to the overall performance of a call center. There is only one event type in this category:

- Call Center Monitoring

6.9.1 Call Center Monitoring

An initial `CallCenterMonitoringEvent` is sent upon subscription to the event package. The event is also triggered for the following events:

- An agent is assigned/unassigned from the call center.
- An agent's staffed status changes.
- A staffed agent's idle status changes.
- A staffed agent transitions to/from the *Unavailable* state.

- An initial subscription or refresh.

NOTE: To avoid a large number of events from being sent in a short time period, Cisco BroadWorks implements the throttling of the events for this event package. By default, Cisco BroadWorks sends at most one event every five seconds for a given subscription to this event package. As long as there are queued calls in the Call Center, this event will automatically be generated and transmitted at every throttling interval. If the last call was removed from the queue during the throttling interval, this event will be generated and transmitted at the end of the throttling interval.

This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>monitoringStatus</i>	CallCenterMonitoringStatus	Yes	The current call center monitoring status

The following table lists which parameters the CallCenterMonitoringStatus are mandatory or optional.

Parameter	Mandatory or Optional	Comments
<i>averageHandlingTime</i>	Mandatory	
<i>expectedWaitTime</i>	Mandatory	
<i>averageSpeedOfAnswer</i>	Mandatory	
<i>longestWaitTime</i>	Mandatory	
<i>numCallsInQueue</i>	Mandatory	
<i>numAgentsAssigned</i>	Mandatory	
<i>numAgentsStaffed</i>	Mandatory	
<i>numStaffedAgentsIdle</i>	Mandatory	
<i>numStaffedAgentsUnavailable</i>	Mandatory	

6.10 Call Park Events

A Call Park event is an event that reports on calls that are parked against a particular subscriber. There is only one event type in this category:

- Call Parked Against

6.10.1 Call Parked Against

The *CallParkedAgainst* event is sent when a call is parked against the target subscriber or when a call is retrieved. It is also sent after a subscription is added or refreshed

This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>parkedParty</i>	PartyInformation	No	<p>Contains the details of the parked party.</p> <ul style="list-style-type: none"> ▪ Present when a call is parked against the user. ▪ Not present when a call is no longer parked against the user.

6.11 Meet-Me Conference Events (**AS only**)

A Meet-Me conference event is an event that reports a change associated to a Meet-Me conference.

Remote applications can subscribe to events associated to a specific conference by providing a targetId (equal to the user ID of the conference bridge) and a targetSubId (equal to the conference ID). See AddUserSubIdSubscriptionRequest in section [5.11.2 Add SubUser Subscription](#) for details on the request to create such a subscription. For this type of subscription:

- An initial subscription event is issued and includes the current state of the conference (if it has started) along with the participant list.
- Only events specific to the subscribed conference are generated. No event is sent for the other conferences on this bridge.

If the subscribing application provides the user ID of the conference bridge virtual subscriber without specifying the conference ID of the conference (in other words, if it subscribes using the AddUserSubscriptionRequest), then events are generated for all conferences on the bridge.

No initial subscription event is issued for a subscription that does not specify the conference ID (for example, user ID with no subID, group, and so on).

The next subsections provide detailed information for all the available Meet-Me conference event types:

- Conference Started
- Conference Stopped
- Conference Locked
- Conference Unlocked
- Conference Recording Started
- Conference Recording Stopped
- Conference Recording Paused
- Conference Recording Resumed
- Conference Lecture Mode Started
- Conference Lecture Mode Stopped
- Conference Auto Lecture Mode Started
- Conference Participant Joined
- Conference Participant Left
- Conference Participant Muted
- Conference Participant Unmuted

- Conference Participant Held
- Conference Participant Retrieved
- Conference Participant Updated
- Conference Participant Outdial Initiated
- Conference Participant Outdial Declined
- Conference Subscription Event
- Conference Subscription Resync Event
- Conference Security Classification Updated
- Conference Client Session Info Updated

6.11.1 Meet-Me Conference Started

The *Meet-Me Conference Started* event is produced when a Meet-Me conference starts. A conference starts when the first participant joins unless the *Moderator required to start conference* option is selected for the conference, in this case the conference does not start until a moderator joins the conference.

Note that conferences can be started more than once, for example, a reservationless conference can be instantiated many times. Each time a conference starts, it initializes the conference information according to the configuration of the conference as well as its participant list according to the current participants in the conference.

It is possible for a conference to require a moderator before starting. In this case, any non-moderators who connect prior to the moderator joining are placed on hold. Only when a moderator joins is an event sent followed by a join event containing a list of the participants who have now entered the call, including the moderator.

6.11.1.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `MeetMeConferenceStartedEvent`. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>securityClassification</code>	<code>NonEmptyToken</code>	No	The conference call classification level. The classification is determined by selecting the lowest current classification level from all participant(s). Present if the conference owner has the Security Classification service assigned.

6.11.2 Meet-Me Conference Stopped

The *Conference Stopped* event is produced when a Meet-Me conference stops. A conference stops either when the last participant has left, or when the last moderator has left, in the case of a conference, which is set to end when moderator departs.

Once the `MeetMeConferenceStoppedEvent` has been issued, no further `MeetMeParticipantLeftEvent` events are sent, as the implication is that all participants have left.



6.11.2.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *MeetMeConferenceStoppedEvent*. This event contains only the common parameters listed at the beginning of this section.

6.11.3 Meet-Me Conference Locked

The *Meet-Me Conference Locked* event is generated when a Meet-Me conference is locked. The event indicates the user who has locked the conference.

6.11.3.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *MeetMeConferenceLockedEvent*. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>participant</i>	MeetMeConferenceParticipant	No	Information on the participant that locked the conference. Not present if the event is the result of a CTI request.

The following table provides the list of parameters used in the participant by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>partyInformation</i>	Mandatory	
<i>activeTalker</i>	Optional	Present when the participant is the active talker on the conference.
<i>held</i>	Optional	Present when the participant has been put on hold by a host.
<i>moderator</i>	Optional	Present when the participant is a moderator in the conference.
<i>muted</i>	Optional	Present when the participant has been muted by a host.
<i>uniqueIdentifier</i>	Optional	Present when the participant has keyed a self-identifying pin while joining the conference.

6.11.4 Meet-Me Conference Unlocked

The *Meet-Me Conference Unlocked* event is generated when a Meet-Me conference is unlocked. The event indicates the user who has unlocked the conference.

6.11.4.1 Event Type Definition

Cisco BroadWorks issues this event by sending a MeetMeConferenceUnlockedEvent. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>participant</i>	MeetMeConferenceParticipant	No	Information on the participant that unlocked the conference. Not present if the event is the result of a CTI request.

The following table provides the list of parameters used in the participant by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>partyInformation</i>	Mandatory	
<i>activeTalker</i>	Optional	Present when the participant is the active talker on the conference.
<i>held</i>	Optional	Present when the participant has been put on hold by a host.
<i>moderator</i>	Optional	Present when the participant is a moderator in the conference.
<i>muted</i>	Optional	Present when the participant has been muted by a host.
<i>uniqueIdentifier</i>	Optional	Present when the participant has keyed a self-identifying pin while joining the conference.

6.11.5 Meet-Me Conference Recording Started

A *Meet-Me Conference Recording Started* event is generated when a Meet-Me conference's recording service is started for the first time, or having previously been stopped.

6.11.5.1 Event Type Definition

Cisco BroadWorks issues this event by sending a MeetMeConferenceRecordingStartedEvent. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>participant</i>	MeetMeConferenceParticipant	No	Information on the participant that started the recording. Not present if the event is the result of a CTI request.



The following table provides the list of parameters used in the participant by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>partyInformation</i>	Mandatory	
<i>activeTalker</i>	Optional	Present when the participant is the active talker on the conference.
<i>held</i>	Optional	Present when the participant has been put on hold by a host.
<i>moderator</i>	Optional	Present when the participant is a moderator in the conference.
<i>muted</i>	Optional	Present when the participant has been muted by a host.

6.11.6 Meet-Me Recording Stopped

A *Meet-Me Conference Recording Stopped* event is generated when a Meet-Me conference's recording service is stopped.

If conference recording is re-initiated after having been stopped, a *Conference Recording Started* event is generated (and not a *Conference Recording Resumed* event).

6.11.6.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `MeetMeConferenceRecordingStoppedEvent`. This event contains only the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>participant</i>	<code>MeetMeConferenceParticipant</code>	No	Information on the participant that stopped the recording. Not present if the event is the result of a CTI request.
<i>reason</i>	<code>MeetMeConferenceRecordingStoppedReason</code>	Yes	Indicates why the recording has stopped.

The following table provides the list of parameters used in the participant by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>partyInformation</i>	Mandatory	
<i>activeTalker</i>	Optional	Present when the participant is the active talker on the conference.
<i>held</i>	Optional	Present when the participant has been put on hold by a host.
<i>moderator</i>	Optional	Present when the participant is a moderator in the conference.
<i>muted</i>	Optional	Present when the participant has been muted by a host.

6.11.7 Meet-Me Recording Paused

A *Meet-Me Conference Recording Paused* event is generated when a Meet-Me conference's recording service is paused.

6.11.7.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `MeetMeConferenceRecordingPausedEvent`. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>participant</code>	<code>MeetMeConferenceParticipant</code>	No	Information on the participant that paused the recording. Not present if the event is the result of a CTI request.

The following table provides the list of parameters used in the participant by this event.

Parameter	Mandatory or Optional	Comments
<code>callId</code>	Mandatory	
<code>partyInformation</code>	Mandatory	
<code>activeTalker</code>	Optional	Present when the participant is the active talker on the conference.
<code>held</code>	Optional	Present when the participant has been put on hold by a host.
<code>moderator</code>	Optional	Present when the participant is a moderator in the conference.
<code>muted</code>	Optional	Present when the participant has been muted by a host.
<code>uniqueIdentifier</code>	Optional	Present when the participant has keyed a self-identifying pin while joining the conference.

6.11.8 Meet-Me Recording Resumed

A *Meet-Me Conference Recording Resumed* event is generated when a Meet-Me conference recording service is resumed, having previously been paused.

6.11.8.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `MeetMeConferenceRecordingResumedEvent`. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>participant</code>	<code>MeetMeConferenceParticipant</code>	No	Information on the participant that resumed the recording. Not present if the event is the result of a CTI request.

The following table provides the list of parameters used in the participant by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>partyInformation</i>	Mandatory	
<i>activeTalker</i>	Optional	Present when the participant is the active talker on the conference.
<i>held</i>	Optional	Present when the participant has been put on hold by a host.
<i>moderator</i>	Optional	Present when the participant is a moderator in the conference.
<i>muted</i>	Optional	Present when the participant has been muted by a host.
<i>uniqueIdentifier</i>	Optional	Present when the participant has keyed a self-identifying pin while joining the conference.

6.11.9 Meet-Me Lecture Mode Started

A *Meet-Me Conference Lecture Mode Started* event is generated when a Meet-Me conference's lecture mode is started.

The lecture mode indicates that all conference participants are muted apart from the moderators of the conference. There is no additional *MeetMeParticipantMutedEvent* event sent for the muting of the participants when they are muted via lecture mode.

6.11.9.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *MeetMeConferenceLectureModeStartedEvent*. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>participant</i>	<i>MeetMeConferenceParticipant</i>	No	Information on the participant that started the lecture mode. Not present if the event is the result of a CTI request.

The following table provides the list of parameters used in the participant by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>partyInformation</i>	Mandatory	
<i>activeTalker</i>	Optional	Present when the participant is the active talker on the conference.
<i>held</i>	Optional	Present when the participant has been put on hold by a host.
<i>moderator</i>	Optional	Present when the participant is a moderator in the conference.
<i>muted</i>	Optional	Present when the participant has been muted by a host.

Parameter	Mandatory or Optional	Comments
<i>uniqueIdentifier</i>	Optional	Present when the participant has keyed a self-identifying pin while joining the conference.

6.11.10 Meet-Me Lecture Mode Stopped

A *Meet-Me Conference Lecture Mode Stopped* event is generated when a Meet-Me conference lecture mode is stopped.

6.11.10.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *MeetMeConferenceLectureModeStoppedEvent*. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>participant</i>	<i>MeetMeConferenceParticipant</i>	No	Information on the participant that stopped the lecture mode. Not present if the event is the result of a CTI request.

The following table provides the list of parameters used in the participant by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>partyInformation</i>	Mandatory	
<i>activeTalker</i>	Optional	Present when the participant is the active talker on the conference.
<i>held</i>	Optional	Present when the participant has been put on hold by a host.
<i>moderator</i>	Optional	Present when the participant is a moderator in the conference.
<i>muted</i>	Optional	Present when the participant has been muted by a host.
<i>uniqueIdentifier</i>	Optional	Present when the participant has keyed a self-identifying pin while joining the conference.

6.11.11 Meet-Me Auto Lecture Mode Started

A *Meet-Me Conference Auto Lecture Mode Started* event is generated when the automatic lecture mode is started.

When a conference is transitioned from Lecture Mode to Automatic Lecture Mode, all the non-moderator participants who are in an unmuted state remain unmuted even after the Automatic Lecture Mode is triggered. This message includes the optional list of (non-moderator) participants who should remain unmuted when applicable.



6.11.11.1 Event Type Definition

Cisco BroadWorks issues this event by sending a MeetMeConferenceAutoLectureModeStartedEvent. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>unmutedParticipant</i>	MeetMeConferenceParticipant	No	This is the list of non-moderator conference participants who should remain unmuted once Automatic Lecture Mode has been entered. All other non-moderator participants are muted.

The following table provides the list of parameters used in the participant by this event.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>partyInformation</i>	Mandatory	
<i>activeTalker</i>	Optional	Present when the participant is the active talker on the conference.
<i>held</i>	Optional	Present when the participant has been put on hold by a host.
<i>muted</i>	Optional	Present when the participant has been muted by a host.
<i>uniqueIdentifier</i>	Optional	Present when the participant has keyed a self-identifying pin while joining the conference.

6.11.12 Meet-Me Participant Joined

A *Meet-Me Participant Joined* event is generated when one or more participants join a Meet-Me conference.

6.11.12.1 Event Type Definition

Cisco BroadWorks issues this event by sending a MeetMeConferenceParticipantJoinedEvent. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>participants</i>	MeetMeConferenceParticipants	Yes	The list of conference participants joining the conference.

The following table provides the list of participant parameters used by each entry in the list.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>partyInformation</i>	Mandatory	

Parameter	Mandatory or Optional	Comments
<i>activeTalker</i>	Optional	Present when the participant is the active talker on the conference.
<i>held</i>	Optional	Present when the participant has been put on hold by a host.
<i>moderator</i>	Optional	Present when the participant is a moderator in the conference.
<i>muted</i>	Optional	Present when the participant has been muted by a host.
<i>uniqueIdentifier</i>	Optional	Present when the participant has keyed a self-identifying pin while joining the conference.

6.11.13 Meet-Me Participant Left

A *Meet-Me Participant Left* event is generated when one or more participants leave a Meet-Me conference.

6.11.13.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `MeetMeConferenceParticipantLeftEvent`. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>callIds</i>	<code>MeetMeConferenceCallIdList</code>	No	The list of conference participants leaving the conference. Not present if the event is the result of a CTI request.

The following table provides the list of `callId` used by each entry in the list.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	

6.11.14 Meet-Me Participant Muted

A *Meet-Me Participant Muted* event is generated when one or more participants are muted in a Meet-Me conference.

6.11.14.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `MeetMeParticipantMutedEvent`. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>conferenceParticipants</i>	<code>MeetMeConferenceParticipants</code>	Yes	The list of conference participants being muted.



The following table provides the list of participant parameters used by each entry in the list.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>partyInformation</i>	Mandatory	
<i>activeTalker</i>	Optional	Present when the participant is the active talker on the conference.
<i>held</i>	Optional	Present when the participant has been put on hold by a host.
<i>moderator</i>	Optional	Present when the participant is a moderator in the conference.
<i>muted</i>	Mandatory	
<i>uniqueIdentifier</i>	Optional	Present when the participant has keyed a self-identifying pin while joining the conference.

6.11.15 Meet-Me Participant Unmuted

A *Meet-Me Participant Unmuted* event is generated when one or more participants are unmuted in a Meet-Me conference.

6.11.15.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *MeetMeParticipantUnmutedEvent*. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>conferenceParticipants</i>	MeetMeConferenceParticipants	Yes	The list of conference participants being unmuted.

The following table provides the list of participant parameters used by each entry in the list.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>partyInformation</i>	Mandatory	
<i>activeTalker</i>	Optional	Present when the participant is the active talker on the conference.
<i>held</i>	Optional	Present when the participant has been put on hold by a host.
<i>moderator</i>	Optional	Present when the participant is a moderator in the conference.
<i>uniqueIdentifier</i>	Optional	Present when the participant has keyed a self-identifying pin while joining the conference.

6.11.16 Meet-Me Participant Held

A *Meet-Me Participant Held* event is generated when one or more participants are being held in a Meet-Me conference.

6.11.16.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `MeetMeParticipantHeldEvent`. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>conferenceParticipants</code>	<code>MeetMeConferenceParticipants</code>	Yes	The list of conference participants being held.

The following table provides the list of participant parameters used by each entry in the list.

Parameter	Mandatory or Optional	Comments
<code>callId</code>	Mandatory	
<code>partyInformation</code>	Mandatory	
<code>activeTalker</code>	Optional	Present when the participant is the active talker on the conference.
<code>held</code>	Mandatory	
<code>moderator</code>	Optional	Present when the participant is a moderator in the conference.
<code>muted</code>	Mandatory	Present when the participant has been muted by a host.
<code>uniqueIdentifier</code>	Optional	Present when the participant has keyed a self-identifying pin while joining the conference.

6.11.17 Meet-Me Participant Retrieved

A *Meet-Me Participant Retrieved* event is generated when one or more participants are unheld in a Meet-Me conference.

6.11.17.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `MeetMeParticipantRetrievedEvent`. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>conferenceParticipants</code>	<code>MeetMeConferenceParticipants</code>	Yes	The list of conference participants being retrieved.



The following table provides the list of participant parameters used by each entry in the list.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>partyInformation</i>	Mandatory	
<i>activeTalker</i>	Optional	Present when the participant is the active talker on the conference.
<i>moderator</i>	Optional	Present when the participant is a moderator in the conference.
<i>muted</i>	Mandatory	Present when the participant has been muted by a host.
<i>uniqueIdentifier</i>	Optional	Present when the participant has keyed a self-identifying pin while joining the conference.

6.11.18 Meet-Me Participant Updated

A *Meet-Me Participant Updated* event is generated when one or more participants have their participant information updated in a Meet-Me conference. This even is generated:

- When a participant that has already joined the conference chooses to log in as a moderator,
- When the party information for the participant changes due for example to a call transfer,
- When a participant has entered or changed his Meet-Me conferencing unique identifier, or
- When a participant becomes the new active talker, or is no longer the active talker.

6.11.18.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *MeetMeParticipantUpdatedEvent*. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>conferenceParticipants</i>	MeetMeConferenceParticipants	Yes	The list of conference participants being updated.

The following table provides the list of participant parameters used by each entry in the list.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>partyInformation</i>	Mandatory	
<i>activeTalker</i>	Optional	Present when the participant is the active talker on the conference.
<i>held</i>	Optional	Present when the participant has been put on hold by a host.
<i>moderator</i>	Optional	Present when the participant is a moderator in the conference.

Parameter	Mandatory or Optional	Comments
<i>muted</i>	Optional	Present when the participant has been muted by a host.
<i>uniqueIdentifier</i>	Optional	Present when the participant has keyed a self-identifying pin while joining the conference.

6.11.19 Meet-Me Conference Subscription Event and Meet-Me Conference Resync Subscription Event

A *Meet-Me Conference Subscription* event is issued immediately after a subscription against the Meet-Me Conference event package is added or refreshed, provided that the subscription target is a Meet-Me conference. A remote application can subscribe against a Meet-Me conference target by setting the subscription targetId to the conference bridge virtual subscriber and the targetSubId to the conference ID. A *Meet-Me Conference Subscription* event is not issued when the target is a conference bridge or a collection of subscribers (for example, a group).

The *Meet-Me Conference Subscription Resync* event is generated when the Application Server shuts down abnormally allowing a remote application to resynchronize.

Both the *Meet-Me Conference Subscription* event and the *Meet-Me Conference Subscription Resync* event share the same definition.

6.11.19.1 Event Type Definition

Cisco BroadWorks issues this event by sending a MeetMeConferenceSubscriptionEvent (or a MeetMeConferenceSubscriptionResyncEvent). This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>conference</i>	MeetMeConference	No	Present if the conference has started.

The following table provides the list of parameters used in the conference by this event.

Parameter	Mandatory or Optional	Comments
<i>conferenceInfo</i>	Mandatory	
<i>conferenceParticipants</i>	Mandatory	

6.11.20 Meet-Me Conference Security Classification Updated

A *Meet-Me Conference Security Classification Updated* event is generated when there is a change to the security classification level of a Meet-Me conference.

This event is generated if the conference owner has the Security Classification service assigned.

The event contains the classification level name and classification priority determined by selecting the lowest-ranked *user classification level* from all participant(s).

The event is generated when there is a change to the conference call classification level, triggered when:

- A participant joins the conference.

- A participant leaves the conference.
- A participant changes their current classification level.
- A participant's connected identity changes in complex call scenarios (that is, transfers).

This event is **not** generated when the conference starts. When the conference starts, the security classification details are provided in the *MeetMeConferenceStartedEvent*.

6.11.20.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *MeetMeConferenceSecurityClassificationUpdated*. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>securityClassification</i>	NonEmptyToken	Yes	The conference security classification value. Determined by selecting the lowest-ranked current classification level from all participant(s).

6.11.21 Meet-Me Conference Client Session Info Updated

A *Meet-Me Conference Client Session Info Updated* event is generated when there is a change there are changes to the client session info for the conference.

6.11.21.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *MeetMeConferenceClientSessionInfoUpdatedEvent*. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>conferenceinfo</i>	MeetMeConferenceInfo	Yes	The client session info.

6.12 Collaborate Room Events (**AS only**)

A Collaborate Room event is an event that reports a change associated to a Collaborate Room.

Remote applications can subscribe to events associated to a specific room by providing a *targetId* (equal to the owner of the room) and a *targetSubId* (equal to the room ID). See *AddUserSubIdSubscriptionRequest* in section [5.11.2 Add SubUser Subscription](#) for details on the request to create such a subscription. For this type of subscription:

- An initial subscription event is issued which includes the current state of the room (if it has started) along with the participant list.
- Only events specific to the subscribed conference are generated. No event is sent for the other rooms owned by the subscriber.

If the subscribing application provides the user ID of the Collaborate subscriber without specifying the room ID (in other words, if it subscribes using the *AddUserSubscriptionRequest*), then events are generated for all rooms owned by the subscriber.

No initial subscription event is issued for a subscription that does not specify the room ID (for example, user ID with no subID, group, and so on).



The next subsections provide detailed information for all the available Collaborate Room event types:

- Collaborate Room Started
- Collaborate Room Stopped
- Collaborate Room Locked
- Collaborate Room Unlocked
- Collaborate Room Participant Joined
- Collaborate Room Participant Left
- Collaborate Room Participant Muted
- Collaborate Room Participant Unmuted
- Collaborate Room Participant Updated
- Collaborate Room Subscription Event
- Collaborate Room Subscription Resync Event
- Collaborate Room Client Session Info Updated

6.12.1 Collaborate Room Started

The *Collaborate Room Started Event* is generated when a Collaborate Room conference starts. A room starts when the first participant joins unless the *owner required to start room* option is selected for the room, in which case the room does not start until an owner joins the room.

Note that rooms can be started more than once. Each time a room starts, it initializes the room information according to the configuration of the room.

It is possible for a conference to require an owner before starting. In this case, any other participant who connects prior to the owner joining is placed on hold. Only when an owner joins is a *CollaborateRoomStartedEvent* sent. It is followed by a join event that contains a list of the participants who have now entered the call, including the owner.

6.12.1.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *CollaborateRoomStartedEvent*. This event contains only the common parameters listed at the beginning of this section.

6.12.2 Collaborate Room Stopped

The *Collaborate Room Stopped Event* is generated when a Collaborate Room conference stops.

For a room that is set to end when the owner departs, when the owner leaves, the room stops. For a room that does not have this restriction, the room stops when the last participant or owner leaves.

Once the *CollaborateRoomStoppedEvent* has been issued, no further *CollaborateRoomLeftEvent* events are sent, as the implication is that all participants have left.

6.12.2.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *CollaborateRoomStoppedEvent*. This event contains only the common parameters listed at the beginning of this section.



6.12.3 Collaborate Room Locked

The *Collaborate Room Locked* event is generated when a Collaborate Room is locked.

6.12.3.1 Event Type Definition

Cisco BroadWorks issues this event by sending a CollaborateRoomLockedEvent. This event contains only the common parameters listed at the beginning of this section.

6.12.4 Collaborate Room Unlocked

The *Collaborate Room Unlocked* event is generated when a Collaborate Room is unlocked.

6.12.4.1 Event Type Definition

Cisco BroadWorks issues this event by sending a CollaborateRoomUnlockedEvent. This event contains only the common parameters listed at the beginning of this section.

6.12.5 Collaborate Room Participant Joined

A *Collaborate Room Participant Joined* event is generated when one or more participants join a Collaborate Room.

6.12.5.1 Event Type Definition

Cisco BroadWorks issues this event by sending a CollaborateRoomParticipantJoinedEvent. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>collaborateRoomParticipants</i>	CollaborateRoomParticipants	Yes	The list of conference participants joining the conference.

The following table provides the list of participant parameters used by each entry in the list.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>partyInformation</i>	Mandatory	
<i>activeTalker</i>	Optional	Present when the participant is the active talker in the room.
<i>owner</i>	Optional	Present when the participant is the owner or the room.
<i>muted</i>	Mandatory	Present when the participant has been muted.

6.12.6 Collaborate Room Participant Left

A *Collaborate Room Left* event is generated when one or more participants leave a Collaborate Room.

6.12.6.1 Event Type Definition

Cisco BroadWorks issues this event by sending a CollaborateRoomParticipantLeftEvent. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>callIds</i>	CollaborateRoomCallIdList	No	The list of conference participants leaving the conference.

The following table provides the list of callId used by each entry in the list.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	

6.12.7 Collaborate Room Participant Muted

A *Collaborate Room Participant Muted* event is generated when one or more participants are muted in a Collaborate Room.

6.12.7.1 Event Type Definition

Cisco BroadWorks issues this event by sending a CollaborateRoomParticipantMutedEvent. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>collaborateRoomParticipants</i>	CollaborateRoomParticipants	Yes	The list of room participants being muted.

The following table provides the list of participant parameters used by each entry in the list.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>partyInformation</i>	Mandatory	
<i>activeTalker</i>	Optional	Present when the participant is the active talker in the room.
<i>owner</i>	Optional	Present when the participant is the owner or the room.
<i>muted</i>	Mandatory	Present when the participant has been muted.

6.12.8 Collaborate Room Participant Unmuted

A *Collaborate Room Participant Unmuted* event is generated when one or more participants are unmuted in a Collaborate Room.

6.12.8.1 Event Type Definition

Cisco BroadWorks issues this event by sending a CollaborateRoomUnmutedEvent. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>collaborateRoomParticipants</i>	CollaborateRoomParticipants	Yes	The list of room participants being unmuted.

The following table provides the list of participant parameters used by each entry in the list.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>partyInformation</i>	Mandatory	
<i>activeTalker</i>	Optional	Present when the participant is the active talker in the room.
<i>owner</i>	Optional	Present when the participant is the owner or the room.

6.12.9 Collaborate Room Participant Updated

A *Collaborate Room Participant Updated* event is generated when one or more participants have their participant information updated in a Collaborate Room. This even is generated:

- When the party information for the participant changes due for example to a call transfer, or
- When a participant becomes the new active talker or is no longer the active talker.

6.12.9.1 Event Type Definition

Cisco BroadWorks issues this event by sending a CollaborateRoomParticipantUpdatedEvent. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>collaborateRoomParticipants</i>	CollaborateRoomParticipants	Yes	The list of room participants being updated.

The following table provides the list of participant parameters used by each entry in the list.

Parameter	Mandatory or Optional	Comments
<i>callId</i>	Mandatory	
<i>partyInformation</i>	Mandatory	

Parameter	Mandatory or Optional	Comments
<i>activeTalker</i>	Optional	Present when the participant is the active talker in the room.
<i>owner</i>	Optional	Present when the participant is the owner or the room.
<i>muted</i>	Optional	Present when the participant has been muted.

6.12.10 Collaborate Room Subscription Event and Collaborate Room Resync Subscription Event

A *Collaborate Room Subscription* event is issued immediately after a subscription against the *Collaborate Room* event package is added or refreshed, provided that the subscription target is a *Collaborate Room*. A remote application can subscribe against a *Collaborate Room* target by setting the subscription targetId to the room owner and the targetSubId to the room ID. A *Collaborate Room Subscription* event is not issued when the target is strictly the owner or a collection of subscribers (for example, a group).

The *Collaborate Room Subscription Resync* event is generated when the Application Server shuts down abnormally allowing a remote application to resynchronize.

Both the *Collaborate Room Subscription* event and the *Collaborate Room Subscription Resync* event share the same definition.

6.12.10.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *CollaborateRoomSubscriptionEvent* (or a *CollaborateRoomSubscriptionResyncEvent*). This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>conference</i>	<i>CollaborateRoom</i>	No	Present if the room is active and started.

The following table provides the list of parameters used in the room by this event.

Parameter	Mandatory or Optional	Comments
<i>roomSessionInfo</i>	Mandatory	
<i>roomParticipants</i>	Mandatory	

6.12.11 Collaborate Room Client Session Info Updated

A *Collaborate Room Client Session Info Updated* event is generated when there are changes to the client session info for the conference.

6.12.11.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `CollaborateRoomClientSessionInfoUpdatedEvent`. This event contains only the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>roomSessionInfo</code>	<code>CollaborateRoomSessionInfo</code>	Yes	The room information, including the client session info value.

6.13 Service Management Events

A service management event is an event that reports a change associated to a service configuration. The next subsections provide detailed information for all the available service management event types:

- Call Forwarding Always
- Call Forwarding No Answer
- Call Forwarding Busy
- Hoteling Guest
- Voice Mail Message Summary

6.13.1 Call Forwarding Always

The *Call Forwarding Always* event is generated when the Call Forwarding Always (CFA) service configuration is modified or after a subscription is added or refreshed. The CFA configuration can be changed through:

- A Modify Call Forwarding Always request
- FAC code (for example, *72)
- Cisco BroadWorks Web portal

6.13.1.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `CallForwardingAlwaysEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>info</code>	<code>CallForwardingAlways</code>	Yes	The new CFA configuration values for this subscriber.

The following table lists which parameters the `CallForwardingAlways` are mandatory or optional.

Parameter	Mandatory or Optional	Comments
<code>isActive</code>	Mandatory	
<code>forwardToPhoneNumber</code>	Optional	Must be present if <code>isActive</code> is set to "true".

Parameter	Mandatory or Optional	Comments
<i>ringSplash</i>	Mandatory	

6.13.2 Call Forwarding No Answer

The *Call Forwarding No Answer* event is generated when the Call Forwarding No Answer (CFNA) service configuration is modified or after a subscription is added or refreshed. The CFNA configuration can be changed through:

- A Modify Call Forwarding No Answer request
- Feature access code (for example, *92)
- Cisco BroadWorks web portal

6.13.2.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `CallForwardingNoAnswerEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>info</code>	<code>CallForwardingNoAnswer</code>	Yes	The new CFNA configuration values for this subscriber.

The following table identifies which parameters of the `CallForwardingNoAnswer` are mandatory or optional.

Parameter	Mandatory or Optional	Comments
<code>isActive</code>	Mandatory	
<code>forwardToPhoneNumber</code>	Optional	Must be present if <code>isActive</code> is set to "true".
<code>numberOfRings</code>	Mandatory	

6.13.3 Call Forwarding Busy

The *Call Forwarding Busy* event is generated when the Call Forwarding Busy (CFB) service configuration is modified or after a subscription is added or refreshed. The CFB configuration can be changed through:

- a Modify Call Forwarding Busy request
- FAC code (for example, *90)
- Cisco BroadWorks Web portal

6.13.3.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `CallForwardingBusyEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>info</code>	<code>CallForwardingBusy</code>	Yes	The new CFB configuration values for this subscriber.



The following table identifies which parameters of the *CallForwardingBusy* event are mandatory or optional.

Parameter	Mandatory or Optional	Comments
<i>isActive</i>	Mandatory	
<i>forwardToPhoneNumber</i>	Optional	Must be present if <i>isActive</i> is set to "true".

6.13.4 Do Not Disturb

The *Do Not Disturb* event is generated when Do Not Disturb (DND) service configuration is modified or after a subscription is added or refreshed. The DND configuration can be changed through:

- a Modify Do Not Disturb ding Always request
- FAC code
- Cisco BroadWorks Web portal

6.13.4.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *DoNotDisturbEvent*. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>info</i>	<i>DoNotDisturb</i>	Yes	The new DND configuration values for this subscriber.

The following table identifies which parameters of the *DoNotDisturb* event are mandatory or optional.

Parameter	Mandatory or Optional	Comments
<i>isActive</i>	Mandatory	
<i>ringSplash</i>	Mandatory	

6.13.5 Call Transfer

The Call Transfer event is generated when the Call Transfer service configuration is modified or after a subscription is added or refreshed.

6.13.5.1 Event Type Definition

Cisco BroadWorks issues this event by sending a *CallTransfer* event. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>unassigned</i>	<i>EmptyContent</i>	No	When present, indicates that service is unassigned.
<i>info</i>	<i>CallTransferInfo</i>	Yes	Notification data for the Call Transfer service. Not present if the service is unassigned. For more information, see section 8 Type Definitions .

The following table provides the list of info parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>recallActive</i>	Mandatory	
<i>recallNumberOfRings</i>	Mandatory	
<i>busyCampOnActive</i>	Mandatory	
<i>useDiversionInhibitorForBlindTransfer</i>	Mandatory	
<i>useDiversionInhibitorForConsultativeCalls</i>	Mandatory	

6.13.6 Call Waiting

The *Call Waiting* event is generated when the Call Waiting service configuration is modified or after a subscription is added or refreshed.

6.13.6.1 Event Type Definition

Cisco BroadWorks issues this event by sending a Call Waiting event. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>unassigned</i>	EmptyContent	No	When present, indicates that service is unassigned.
<i>Info</i>	ActivatableServiceInfo	No	Present when the service is assigned.

The following table provides the info parameter used by this event.

Parameter	Mandatory or Optional	Comments
<i>active</i>	Mandatory	

6.13.7 Last Number Redial

The Last Number Redial event is generated when the Last Number Redial service configuration is modified or after a subscription is added or refreshed.

6.13.7.1 Event Type Definition

Cisco BroadWorks issues this event by sending a LastNumberRedialEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>unassigned</i>	EmptyContent	No	When present, indicates that the service is unassigned.

6.13.8 Music On Hold

The Music On Hold event is generated when the Music On Hold service configuration is modified or after a subscription is added or refreshed.

6.13.8.1 Event Type Definition

Cisco BroadWorks issues this event by sending a MusicOnHoldEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>unassigned</i>	EmptyContent	No	When present, indicates that the service is unassigned.

6.13.9 N-Way Call

The N-Way Call event is generated when the N-Way Call Service configuration is modified or after a subscription is added or refreshed.

6.13.9.1 Event Type Definition

Cisco BroadWorks issues this event by sending an N-WayCallEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>unassigned</i>	EmptyContent	No	When present, indicates that the service is unassigned.

6.13.10 Third-Party Voice Mail Support

The Third-Party Voice Mail Support event is generated when the Third-Party Voice Mail Support service configuration is modified or after a subscription is added or refreshed.

6.13.10.1 Event Type Definition

Cisco BroadWorks issues this event by sending a ThirdPartyVoiceMailSupportEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>unassigned</i>	EmptyContent	No	Present when the service is unassigned.
<i>info</i>	ActivatableServiceInfo	No	Present, when the service is assigned.

The following table provides the info parameter used by this event.

Parameter	Mandatory or Optional	Comments
<i>active</i>	Mandatory	

6.13.11 Three-Way Call

The Three-Way Call event is generated when the Three-Way Call Service configuration is modified or after a subscription is added or refreshed.

6.13.11.1 Event Type Definition

Cisco BroadWorks issues this event by sending a ThreeWayCallEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<i>unassigned</i>	EmptyContent	No	When present, indicates that the service is unassigned.

6.13.12 Voice Messaging

The Voice Messaging event is generated when the Voice Messaging service configuration is modified or after a subscription is added or refreshed.

6.13.12.1 Event Type Definition

Cisco BroadWorks issues this event by sending a VoiceMesssagingEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>unassigned</i>	EmptyContent	No	Present when the service is unassigned.
<i>info</i>	ActivatableServiceInfo	No	Present, when the service is assigned.

The following table provides the info parameter used by this event.

Parameter	Mandatory or Optional	Comments
<i>active</i>	Mandatory	

6.13.13 Sequential Ring

The Sequential Ring event is generated when the Sequential Ring service configuration is modified or after a subscription is added or refreshed.

6.13.13.1 Event Type Definition

Cisco BroadWorks issues this event by sending a SequentialRingEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>unassigned</i>	EmptyContent	No	When present, indicates that service is unassigned.
<i>info</i>	SequentialRing	No	Notification data for the Sequential Ring service. Not present if the service is unassigned. For more information, see section 8 Type Definitions .

The following table provides the list of info parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>ringBaseLocationFirst</i>	Mandatory	
<i>baseLocationNumberOfRings</i>	Mandatory	
<i>continuelfBaseLocationIsBusy</i>	Mandatory	
<i>callerMayStopSearch</i>	Mandatory	
<i>location1</i>	Mandatory	
<i>location2</i>	Mandatory	
<i>location3</i>	Mandatory	
<i>location4</i>	Mandatory	
<i>location5</i>	Mandatory	
<i>criteriaActivationList</i>	Mandatory	

6.13.14 Simultaneous Ring Personal

The Simultaneous Ring Personal event is generated when the Simultaneous Ring Personal service configuration is modified or after a subscription is added or refreshed.

6.13.14.1 Event Type Definition

Cisco BroadWorks issues this event by sending a SimultaneousRingPersonalEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>unassigned</i>	EmptyContent	No	When present, indicates that service is unassigned.
<i>info</i>	SimultaneousRingPersonal	No	Notification data for the Simultaneous Ring Personal service. Not present if the service is unassigned. For more information, see section 8 Type Definitions .

The following table provides the list of info parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>active</i>	Mandatory	
<i>incomingCalls</i>	Mandatory	
<i>simRingLocations</i>	Optional	Only present when one location is set.
<i>criteriaActivationList</i>	Optional	Only present when one criterion is set.

6.13.15 Voice Mail Message Summary

The *Voice Mail Message Summary* event is generated when the Voice Mail content is modified or after a subscription is added or refreshed.

6.13.15.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `VoiceMailMessageSummaryEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>messageSummary</code>	<code>MessageSummary</code>	Yes	The new summary of messages for this voice mail.

6.13.16 Hoteling Guest (AS only)

The *Hoteling Guest* event is generated when the Hoteling Guest configuration is modified or after a subscription is added or refreshed. The guest configuration is changed when for example a guest is associated with a host.

6.13.16.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `HotelingGuestEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameter.

Parameter	Parameter Type	Required	Description
<code>info</code>	<code>HotelingGuest</code>	Yes	The new Hoteling Guest configuration values for this subscriber.

The following table identifies which parameters of the *HotelingGuest* event are mandatory or optional.

Parameter	Mandatory or Optional	Comments
<code>isActive</code>	Mandatory	
<code>enableAssociationLimit</code>	Optional	Present when the association limit is enforced.
<code>associationLimitHours</code>	Optional	Present when the timeout value is set.
<code>hostUserId</code>	Optional	Present when the guest is associated with a host.

6.13.17 BroadWorks Anywhere

The BroadWorks Anywhere event is generated when the BroadWorks Anywhere service configuration is modified or after a subscription is added or refreshed.

6.13.17.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `BroadWorksAnywhereEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>unassigned</code>	<code>EmptyContent</code>	No	When present, indicates that service is unassigned.

Parameter	Parameter Type	Required	Description
<i>info</i>	BroadWorksAnywhereInfo	Yes	Notification data for the BroadWorks Anywhere service. Not present if the service is unassigned. For more information, see section 8 Type Definitions .

The following table provides the list of info parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>alertAllLocationsForClickToDialCalls</i>	Mandatory	
<i>locations</i>	Optional	This is present when at least one location is set in the BroadWorks Anywhere service.

6.13.18 Remote Office

The Remote Office event is generated when the Remote Office service configuration is modified or after a subscription is added or refreshed.

6.13.18.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `RemoteOfficeEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>unassigned</i>	EmptyContent	No	When present, indicates that the service is unassigned.
<i>info</i>	RemoteOffice	No	Present, when the service is assigned.

The following table provides the info parameter used by this event.

Parameter	Mandatory or Optional	Comments
<i>remoteOfficeNumber</i>	Optional	Only present when the service is enabled.

6.13.19 CommPilot Express

The CommPilot Express event is generated when the CommPilot Express service configuration is modified or after a subscription is added or refreshed.

6.13.19.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `CommPilotExpressEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>unassigned</i>	EmptyContent	No	Present when the service is unassigned.
<i>info</i>	CommPilotExpress	No	Present, when the service is assigned.



The following table provides the list of info parameters used by this event.

Parameter	Mandatory or Optional	Comments
<code>profile</code>	Mandatory	
<code>availableInOffice</code>	Mandatory	
<code>availableOutOfOffice</code>	Mandatory	
<code>busy</code>	Mandatory	
<code>unavailable</code>	Mandatory	

6.13.20 Flexible Seating

The *Flexible Seating* event is generated when the Flexible Seating Guest configuration is modified or after a subscription is added or refreshed.

6.13.20.1 Event Type Definition

Cisco BroadWorks issues this event by sending a `FlexibleSeatingGuestEvent`. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<code>unassigned</code>	<code>EmptyContent</code>	No	Present when the service is unassigned.
<code>info</code>	<code>FlexibleSeatingGuest</code>	No	Present, when the service is assigned.

The following table provides the list of info parameters used by this event.

Parameter	Mandatory or Optional	Comments
<code>active</code>	Mandatory	
<code>enableAssociationLimit</code>	Mandatory	
<code>associationLimitHours</code>	Mandatory	
<code>unlockPhonePINCode</code>	Optional	Present only if PIN code is set.
<code>accessDevice</code>	Optional	Present only if the guest is associated with the access device.
<code>hostUserId</code>	Optional	Present only if the guest is associated with the host.
<code>hostLastName</code>	N/A	Not present.
<code>hostFirstName</code>	N/A	Not present.
<code>hostAssociationDateTime</code>	N/A	Not present.
<code>hostEnforcesAssociationLimit</code>	N/A	Not present.
<code>hostAssociationLimitHours</code>	N/A	Not present.

6.13.21 Personal Assistant

The *Personal Assistant* event is generated when the Personal Assistant configuration is modified or after a subscription is added or refreshed.

6.13.21.1 Event Type Definition

Cisco BroadWorks issues this event by sending a PersonalAssistantEvent. This event contains the common parameters listed at the beginning of this section and the following specific parameters.

Parameter	Parameter Type	Required	Description
<i>unassigned</i>	EmptyContent	No	Present when the service is unassigned.
<i>info</i>	PersonalAssistant	No	Present when the service is assigned.

The following table provides the list of info parameters used by this event.

Parameter	Mandatory or Optional	Comments
<i>presence</i>	Mandatory	
<i>enableExpirationTime</i>	Mandatory	
<i>expirationTime</i>	Mandatory	

6.14 Subscription Event

A subscription event reports a change associated to a subscription. The only subscription event available is a Subscription Terminated event.

6.14.1 Subscription Terminated

The *SubscriptionTerminated* event is produced when a subscription terminates as described in section [3.4.5.3 Subscription Terminated Event](#).

This event contains no parameter besides the common parameters listed at the beginning of this section.

6.15 Channel Event

A channel event reports a change associated to a channel. The only channel event available is a *Channel Terminated* event.

6.15.1 Channel Terminated

The Channel Terminated event is produced when a channel terminates as described in section [3.4.4.4 Channel Terminated Event](#).



6.15.1.1 Event Type Definition

Cisco BroadWorks issues this event by sending a ChannelTerminatedEvent. This event contains only the following parameters. Note that the remote application does not need to send an event response for this event.

Parameter	Parameter Type	Required	Description
<i>requestId</i>	RequestId	Yes	An identification supplied by Cisco BroadWorks to map the event notification to the client response at the connection layer. Must be returned in the event response.
<i>channelId</i>	ChannelId	Yes	The channel ID for event channel that is being terminated.
<i>reason</i>	ChannelTerminatedReason	Yes	Indicates the reason for channel termination.

7 XML Schema

CTI messages are encoded in XML using the definition provided by the XML schema. This section describes the structure of the XML schema as well as the encoding character sets supported.

7.1 XspXMLInterface Schema

The XspXMLInterface XML schema contains only one file. It provides the base message envelop definitions for all messages carried over the TCP links. The schema defines six global elements as follows:

- request
- response
- event
- eventResponse
- keepalive
- keepaliveResponse

The request, response, event, and eventResponse elements include a reference to a payload element, which defines a substitution group.

In the case of a request message, the payload is substituted with an element of type *RequestPayload*. The CTI schema defines the elements and types that can be used to substitute the payload of a request message.

In the case of response, event, and eventResponse messages, the payload is substituted with elements defined in the Xtended Services Interface schema.

7.2 CTI Schema

The CTI schema defines restrictions on the payload of the request message. The following files are available in the schema zip file:

- *CTISchema.xsd* – This file is the base file for the CTI XML schema files. This file references all included files, including the XspXMLInterface schema and the Xtended Services Interface schema.
- *System/CTISystem.xsd* – This file defines the application controller requests.
- *Events/CTIChannel.xsd* – This file defines the event channel requests.
- *Events/CTISubscription.xsd* – This file defines the event subscription requests.
- *CallSession/CTICallSession.xsd* – This file defines the call and conference requests.
- *Services/CTIServices.xsd* – This file defines the service management requests.
- *ACD/CTIACD.xsd* – This file defines the ACD requests.
- *RoutePoint/CTIRoutePoint.xsd* – This file defines the route point requests.
- *MeetMe/CTIMeetMeConference.xsd* – This file defines the Meet-Me conference requests.
- *Collaborate/CTICollaborateRoom.xsd* – This file defines the Collaborate Room requests.
- *Agent/CTIAgent.xsd* – This file defines the agent requests.



Each request defined in the CTI schema is defined with a type that restricts the RequestPayload type of the XspXmlInterface. The following provides the elements of the RequestPayload type and describes how each element is restricted:

- uri – This element defines the Xtended Services Interface command URI that corresponds to the CTI request. The CTI schema defines a fixed element value that the remote application must use when populating the uri element of a request.

The fixed element value matches the corresponding Xtended Services Interface command URI and includes wildcards for values that are dynamically determined in the URI. For each wildcard, the remote application must include the corresponding value in the params element, which is described in the following list.

The following is an example of a fixed uri element value. The URI contains three wild cards: *subscriberid*, *callid*, and *address*.

```
/com.broadsoft.cti-
actions/v2.0/user/%subscriberid%/calls/%callid%/blindtransfer?%address%
```

- method – This element defines the Xtended Services Interface command method that corresponds to the CTI request. The schema defines a fixed element value that the remote application must include when generating the request. The values can be any of the following: GET DELETE, PUT, and POST.
- version – This element defines the version of the Xtended Services Interface schema that applies for the request being generated.
- params – This element is a sequence of elements that correspond to the wildcards included in the uri element value. An element is included for each wildcard present in the URI. The element tag name corresponds to the wildcard name and the element is defined with a type defined in the Xtended Services Interface schema. In the previous example, the params element is encoded as follows.

```
<params>
<subscriberid>ba3001@virtsanity.mtl.broadsoft.com</subscriberid>
<callid>callhalf-56:0</callid>
<address>tel: 500</address>
</params>
```

- payload – The payload element includes additional input parameters for the request. The schema defines the Xtended Services Interface construct that is required for each CTI request. For CTI requests that do not require additional input parameters, the payload element is empty.

7.2.1 URI Encoding

Because CTI messages are XML-based, they must contain XML-valid data. This means that all CTI elements and attributes must obey XML escaping rules, including the “uri” and “params” elements.

Although the “uri” and “params” elements are URI-like, they must not be percent-encoded. As described in RFC 3986, before applying the XML escaping. An exception to this rule is a plus sign (+) located in a parameter that will end up in the query part of the URI. In that case, the plus sign must be percent-encoded using the lowercase character sequence “%2b”.

Alternatively, if the character sequence “tel” immediately precedes the plus sign, the plus sign does not need to be percent-escaped.



7.3 Xtended Services Interface Schema

The Xtended Services Interface (Xsi) schema defines the types that are used in the payload of the elements defined for the XspXMLInterface.

The following files are available:

- *XSI/Schema.xsd* – This file is the base file for the Xtended Services Interface XML schema files. This file references all included files.
- *XSI/DataTypes.xsd* – This file contains the base definitions for types that are used in the other schema files.
- *XSI/BaseEvent.xsd* – This file contains the base definitions for types that are used in events.
- *System/XSI/System.xsd* – This file contains the types that are used in application controller requests and responses.
- *Events/XSI/Channel.xsd* – This file contains the types that are used in event channel requests and responses.
- *Events/XSI/Subscription.xsd* – This file contains the types that are used in event subscription requests and responses.
- *CallSession/XSI/CallSession.xsd* – This file contains the types that are used in call and conference requests, responses, and events.
- *CallSession/Events/XSI/CallSessionEvents.xsd* – This file contains the events that are sent in the context of a basic call, standard call, or advanced call subscription.
- *Agent/Events/XSI/AgentEvents.xsd* – This file contains the events that are sent in the context of a call center agent subscription.
- *Services/XSI/<XXX>.xsd* – These file contain the types that are used in requests, responses, and events that are related to the various service configuration.
- *Services/Events/XSI/<XXX>.xsd* – These files contain the events that are sent in the context of the various service configuration.
- *Directory/XSI/Directories.xsd* – This file is not used for the CTI interface.
- *Profile/XSI/Profile.xsd* – This file is not used for the CTI interface.
- *ACD/XSI/ACD.xsd* – This file contains the types that are used in ACD requests, responses, and events.
- *ACD/Events/XSI/ACDEvents.xsd* – This file contains the events that are sent in the context of a call center queue subscription.
- *RoutePoint/XSI/RoutePoint.xsd* – This file contains the types that are used in route point requests, responses, and events.
- *RoutePoint/Events/XSI/RoutePointEvents.xsd* – This file contains the events that are sent in the context of a route point queue subscription.
- *Agent/XSI/Agent.xsd* – This file contains the types that are used in agent requests, responses, and events.
- *MeetMeConference/XSI/MeetMeConference.xsd* – This file contains the types that are used in the Meet-Me conference requests, responses, and events.
- *MeetMeConference/Events/XSI/MeetMeConferenceEvents.xsd* – This file contains the events that are sent in the context of a Meet-Me conference subscription.
- *CollaborateRoom/XSI/CollaborateRoom.xsd* – This file contains the types that are used in the Collaborate Room requests, responses, and events.

- *CollaborateRoom /Events/XSI/CollaborateRoomEvents.xsd* – This file contains the events that are sent in the context of a Collaborate Room subscription.
- *Services/Events/XSI/CommPilotExpressEvents.xsd* – This file contains the events that are sent in the context of a CommPilot Express subscription.
- *Services/Events/XSI/HotelingGuestEvents.xsd* – This file contains the events that are sent in the context of a hoteling guest subscription.
- *Services/Events/XSI/VoiceMessagingEvents.xsd* – This file contains the events that are sent in the context of a voice messaging subscription.

7.4 XML Examples

7.4.1 Request

The following provides an example of a conference start request, where the payload element of the request message has been substituted with the ConferenceStartRequest element.

```
<xsp:request version="17.0"
  xmlns:xsp="http://schema.broadsoft.com/XspXMLInterface">
  <requestId>9</requestId>
  <sessionId>MySessionId</sessionId>
  <credentials>YWRtaW5DVE1AYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==</credentials>
  <ConferenceStartRequest>
    <uri>/com.broadsoft.xsi->
  actions/v2.0/user/%subscriberId%/calls/conference</uri>
    <method>POST</method>
    <version/>
    <params>
      <subscriberId>subscriberS1@broadsoft.com</subscriberId>
    </params>
    <xsp:payload>
      <xsi:Conference xmlns:xsi="http://schema.broadsoft.com/xsi">
        <xsi:conferenceParticipantList>
          <xsi:conferenceParticipant>
            <xsi:callId>callhalf-75923:0</xsi:callId>
          </xsi:conferenceParticipant>
          <xsi:conferenceParticipant>
            <xsi:callId>callhalf-75923:1</xsi:callId>
          </xsi:conferenceParticipant>
        </xsi:conferenceParticipantList>
      </xsi:Conference>
    </xsp:payload>
  </ConferenceStartRequest>
</xsp:request>
```

7.4.2 Response

The following provides an example of a response sent by Cisco BroadWorks upon handling a request to add an event channel.

```
<xsp:response version="17.0"
  xmlns:xsp="http://schema.broadsoft.com/XspXMLInterface">
  <requestId>0</requestId>
  <sessionId>1126</sessionId>
  <statusCode>200</statusCode>
  <reason>OK</reason>
  <xsp:payload>
    <Channel xmlns="http://schema.broadsoft.com/xsi">
      <channelId>2dafced4-260e-4d99-96f0-83c11d7e7e4f</channelId>
```

```

<expires>3600</expires>
</Channel>
</xsp:payload>
</xsp:response>

```

7.4.3 Event

The following provides an example of an event message sent by Cisco BroadWorks when a user receives a call.

```

<xsp:event version="17.0"
xmlns:xsp="http://schema.broadsoft.com/XspXMLInterface">
<requestId>0:1967</requestId>
<xsp:payload>
<xsi:Event xsi:type="xsi:SubscriptionEvent"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsi="http://schema.broadsoft.com/xsi">
<xsi:eventID>
a0b5d1d1-a12a-44a6-ad58-1da52a8b30cc</xsi:eventID>
<xsi:sequenceNumber>4</xsi:sequenceNumber>
<xsi:userId>admin@virtsanity.mtl.broadsoft.com</xsi:userId>
<xsi:externalApplicationId>
AppCtlId</xsi:externalApplicationId>
<xsi:subscriptionId>
ceaec03b-a6b7-4681-b4de-287d34cc2b89</xsi:subscriptionId>
<xsi:channelId>
a94890bf-73ae-43d4-87e4-5da75e9a28f7</xsi:channelId>
<xsi:targetId>subscriberS1@broadsoft.com</xsi:targetId>
<xsi:eventData xsi:type="xsi:CallReceivedEvent">
<xsi:call>
<xsi:callId>callhalf-10197:1</xsi:callId>
<xsi:extTrackingId>61:1</xsi:extTrackingId>
<xsi:personality>Terminator</xsi:personality>
<xsi:state>Alerting</xsi:state>
<xsi:remoteParty>
<xsi:name>subscriberS1FirstName
subscriberS1LastName</xsi:name>
<xsi:address>tel:1011</xsi:address>
<xsi:userId>subscriberS1@broadsoft.com</xsi:userId>
<xsi:userDN countryCode="1">
tel:+15006001011;ext=1011</xsi:userDN>
<xsi:callType>Group</xsi:callType>
</xsi:remoteParty>
<xsi:startTime>1271253498397</xsi:startTime>
</xsi:call>
</xsi:eventData>
</xsi:Event>
</xsp:payload>
</xsp:event>

```

7.4.4 EventResponse

The following provides an example of an eventResponse message sent by the remote application in response to an event message.

```

<xsp:eventResponse version="17.0"
xmlns:xsp="http://schema.broadsoft.com/XspXMLInterface">
<requestId>0:1967</requestId>
<xsp:payload>
<xsi:EventResponse xmlns:xsi="http://schema.broadsoft.com/xsi">

```



```
<xsi:eventID>  
a0b5d1d1-a12a-44a6-ad58-1da52a8b30cc</xsi:eventID>  
<xsi:statusCode>200</xsi:statusCode>  
<xsi:reason>OK</xsi:reason>  
</xsi:EventResponse>  
</xsp:payload>  
</xsp:eventResponse>
```

7.4.5 Keepalive

The following provides an example of a keepalive message sent by Cisco BroadWorks.

```
<xsp:keepalive version="17.0"  
xmlns:xsp="http://schema.broadsoft.com/XspXMLInterface">  
    <requestId>15022</requestId>  
    <period>300</period>  
</xsp:keepalive>
```

7.4.6 KeepaliveResponse

The following provides an example of a keepaliveResponse message sent by the remote application in response to a keepalive message.

```
<xsp:keepaliveResponse version="17.0"  
xmlns:xsp="http://schema.broadsoft.com/XspXMLInterface">  
    <requestId>15022</requestId>  
</xsp:keepaliveResponse>
```

7.5 Character Sets Encoding

7.5.1 Incoming Message

Incoming message, that is messages from the remote application to the CTI interface, can be encoded with a character set which do not require Byte Order Marker (BOM). This ensures that the characters of ASCII have their normal positions, width, and values (for example UTF-8, ISO 646, ASCII, some part of ISO 8859, Shift-JIS, EUC, and so on.).

7.5.2 Outgoing Message

Outgoing messages, that is messages from the CTI interface to the remote application, are encoded using the following character sets:

- Service-related responses use the character set determined by the authenticated user's locale.
- Call-related responses use UTF-8.
- Platform-level error responses use the Xtended Services Platform's default character set.
- Events use UTF-8.

8 Type Definitions

This section provides details on the various parameter types used by the CTI interface. It covers the following topics:

- Message parameter type definitions
- Event channel related types
- Event subscription related types
- Common request and response related types
- Common types
- Call related types
- Conference related types
- Service management related types
- Route point related types
- ACD related types
- Agent related types

8.1 Message Parameter Type Definitions

Xtended Services Interface messages are stored in an eXtensible Markup Language (XML) document that must conform to a provided XML schema. This XML document is carried inside requests and responses exchanged between the remote application and Cisco BroadWorks.

Each message exchanged between the remote application and Cisco BroadWorks contains one or multiple parameters. These parameters can be of two types:

- Basic types
- Advanced types

8.1.1 Basic Types

Basic type parameters can be assigned a single value. Examples of basic types are String, Integer, CallId, and so on.

8.1.2 Advanced Types

The advanced types contain multiple parameters. Each parameter can be a basic type or another advanced type. The following table list the main advanced types used by Cisco BroadWorks.

Advanced Parameter Type	Description
Channel	Contains the details of an event channel.
Subscription	Contains the details of an event subscription
Call	Contains the details of a call
Address	Contains information on a phone number address
AccessEndpoint	Contains information on an addressable location terminating a call.
PartyInformation	Contains information on a call session participant.

Advanced Parameter Type	Description
Conference	Contains information on a conference
RoutePointQueueEntry	Contains information about each call maintained at a route point queue.
ACDQueueEntry	Contains information about each call maintained in an ACD.
PlayCollectResult	Contains information about the playCollection result.

Each advanced type is defined using a type definition table. This table provides the following information about each parameter:

- The parameter name
- The parameter type
- Whether any parameter is optional
- The parameter description

For example, if the advanced type A is made of two parameters: one parameter named *b* and one parameter named *c*. The value assigned to parameter *b* is a string. The value assigned to the parameter named *c* is an integer. The parameter named *c* is optional. This means that when a type A parameter is created, it might not have a parameter named *c* in it. The conditions under which *c* is included or not are specific to type A and are documented in the table.

All this information is captured in a type definition table. The following table provides an example of a type definition table for the advanced type A.

Parameter	Parameter Type	Required	Comments
<i>b</i>	String	Yes	Parameter <i>b</i> description.
<i>c</i>	Integer	No	Parameter <i>c</i> description with details of when the parameter should be used since it is optional.

8.1.3 Common Types

8.1.3.1 String

A “String” contains a sequence of characters.

8.1.3.2 Integer

An “Integer” maintains an integer. The range of values that can be assigned to an Integer is from -2147483648 through 2147483647.

8.1.3.3 PositiveInteger

A “PositiveInteger” maintains an integer greater than or equal to 1. The range of a PositiveInteger is from 1 through 2147483647.

8.1.3.4 NonNegativeInteger

A “NonNegativeInteger” maintains an integer greater than or equal to 0. The range of a PositiveInteger is from 0 through 2147483647.

8.1.3.5 Long

A “Long” maintains a long integer. The range of a Long is from –9223372036854775808 through 9223372036854775807.

8.1.3.6 EmptyContent

An “EmptyContent” is a parameter type that has no value assigned. The presence or absence of an *EmptyContent* parameter is enough to provide the required information. This parameter acts like a Boolean value.

8.1.3.7 Boolean

A “Boolean” is a parameter type that can be set to “true” or “false”.

8.1.3.8 Base64Binary

A “base64Binary” represents Base64-encoded arbitrary binary data. The ·value space of base64Binary is the set of finite-length sequences of binary octets. For base64Binary data, the entire binary stream is encoded using the Base64 Alphabet.

8.2 Event Channel

8.2.1 Channel

A Channel maintains information regarding an event channel created for the purpose of event delivery. The following type definition table lists the content of a Channel.

Parameter	Parameter Type	Required	Comments
<i>channelId</i>	ChannelId	No	A system generated unique value for each event channel. The channelId is mandatory while modifying a channel. The ChannelId should not be provided while creation of the channel else ignored.
<i>channelSetId</i>	ChannelSetId	No	This indicates to which event channel set this channel belongs. The ChannelSetId should not be provided while modifying the channel, else ignored.
<i>priority</i>	PositiveInteger	No	The priority for the event channel. If not present, then the event channel has the default priority of 1. The priority should not be provided while modifying the channel, else ignored.
<i>weight</i>	PositiveInteger	No	The weight for the event channel. If not present, then the event channel has the default weight of 500. The priority should not be provided while modifying the channel, else ignored.
<i>expires</i>	PositiveInteger	Yes	The relative expiration time for the event channel in seconds. The expiration is the only element in channel that can be modified after creation of the channel.

Parameter	Parameter Type	Required	Comments
applicationId	ExternalApplicationId	No	The remote application may include the ExternalApplicationId element to enforce the limit of a single channel set per application. If a second channel set is created for the same application, then Cisco BroadWorks terminates all channels associated with the first channel set (effectively terminating the channel set itself).

8.2.1.1 ChannelId

A channelId is a system generated unique value for each event channel. A channelId is represented with a string.

8.2.1.2 ChannelSetId

A ChannelSetId identifies a grouping of event channels. A ChannelSetId is represented with a string.

8.2.1.3 ChannelSet

The ChannelSet type contains a set of Channel.

8.2.2 ChannelResponse

A ChannelResponse maintains summary information regarding an event channel. The following type definition table lists the content of a ChannelResponse.

Parameter	Parameter Type	Required	Comments
channelId	ChannelId	Yes	A system generated unique value for each event channel.
expires	PositiveInteger	Yes	The expiration value is the relative time (delta from the current time) in seconds until the expiration of an event channel.

8.2.3 ChannelTerminatedReason

A ChannelTerminatedReason indicates the cause for channel termination in a Channel Terminated Event.

Reason	Description
DuplicateChannelSet	Cisco BroadWorks limits the creation of a single notification channel set for a user per remote application. A remote application receiving DuplicateChannelSet reason should terminate the current session and inform the user the reason for termination.
UserDeleted	The user for whom the channel was created was deleted.
NonResponsiveChannel	All channels in a channel set are not responding to notification events.
ExplicitDelete	The channel was explicitly deleted either by the remote application or by a Cisco BroadWorks administrative action.
Expired	The channel duration expired.

Reason	Description
UnknownChannel	A notification event included a channel ID that was unknown to the remote application. The remote application replied with an Unknown Channel (404) response.

A ChannelResponse maintains summary information regarding an event channel. The following type definition table lists the content of a ChannelResponse.

Parameter	Parameter Type	Required	Comments
<i>channelId</i>	ChannelId	Yes	A system generated unique value for each event channel.
<i>expires</i>	PositiveInteger	Yes	The expiration value is the relative time (delta from the current time) in seconds until the expiration of an event channel.

8.3 Subscription

8.3.1 Subscription

A Subscription maintains information regarding a subscription. The following type definition table lists the content of a Subscription.

Parameter	Parameter Type	Required	Comments
<i>subscriptionId</i>	SubscriptionId	No	A system generated unique value for each subscription. This should not be provided while creating a new or duplicate (overwriting an existing) subscription.
<i>sharedSubscriptionId</i>	SubscriptionId	No	The unique ID of the shared subscription. Generated by the remote application and sent for subscriptions to the same event package and target ID combination.
<i>subscriberId</i>	UserId	No	The owner of the subscription. This corresponds to the authenticated subscriber. This should not be provided while creating or modifying a subscription.
<i>targetIdType</i>	EntityType	No	The type of target. This should not be provided while creating or modifying a subscription.
<i>targetGroupParentId</i>	String	No	Specify the group parent ID. Only present when the target is a group. This should not be provided while creating or modifying a subscription.
<i>targetId</i>	String	No	The target ID of the Subscription. This consists of a user ID for a targetIdType user, service provider ID/enterprise ID for a targetIdType Enterprise, and group ID for a targetIdType Group. This element is not present for a targetIdType System. This should not be provided while creating or modifying a subscription.

Parameter	Parameter Type	Required	Comments
<i>targetSubId</i>	String	No	Used by specific event packages to provide a second level of granularity to the targetId. For example, a targetUserId for a Meet-Me Conference bridge can have a targetSubId, which identifies a specific conference, which belongs to the bridge. Alternatively, a targetUserId for a Collaborate Room owner can have a targetSubId, which identifies a specific room that belongs to the owner.
<i>event</i>	EventPackage	No	The type of the event package for which the subscription is intended.
<i>expires</i>	Integer	No	The expiration value is the relative time (delta from the current time) in seconds until the expiration of a subscription.
<i>httpContact</i>	HttpContact	No	This parameter is not used by the CTI interface. It is only used when running over HTTP.
<i>channelSetId</i>	String	No	This parameter must be present when using the CTI interface. It is only used when running over TCP.
<i>applicationId</i>	String	No	The application ID of the external application that owns a subscription. This should be globally unique.

8.3.2 SubscriptionId

A SubscriptionId uniquely identifies a subscription. A SubscriptionId is represented with a string.

8.3.3 SubscriptionList

A SubscriptionList contains a list of subscriptions.

8.3.4 EntityType

An EntityType specifies the type of target for a subscription. The following table provides the list of values that can be assigned to an EntityType.

EntityType	Description
User	The subscription target is a user.
Group	The subscription target is a group.
Enterprise	The subscription target is an enterprise.
System	The subscription target is the system.

8.3.5 EventPackage

An EventPackage specifies the name of the event package. The following table provides the list of values that can be assigned to an EventPackage.

Package Name
Advanced Call



Package Name
Basic Call
BroadWorks Anywhere
Call Center Agent
Call Center Agent Monitoring
Call Center Configuration
Call Center Monitoring
Call Center Queue
Call Forwarding Always
Call Forwarding Busy
Call Forwarding No Answer
Call Park
Call Recording
Call Transfer
Call Waiting
CommPilot Express
Do Not Disturb
Flexible Seating
Hoteling Guest
Last Number Redial
Meet Me Conference
Music On Hold
N-Way Call
Remote Office
Route Point Queue
Sequential Ring
Sequential Ring
Simultaneous Ring Personal
Standard Call
Third-Party Voice Mail Support
Three-Way Call
Voice Mail Message Summary
Voice Messaging

8.4 Common Request and Response Related Types

8.4.1 SessionId

A SessionId identifies a session between a remote application and Cisco BroadWorks. A SessionId is represented with a string.

8.4.2 RequestId

A RequestId identifies a request. The requestId is used to map a request to a response. A RequestId is represented with a string.

8.4.3 EventId

An EventId identifies an event. The eventId is used to map an event to a response. An EventId is represented with a string.

8.4.4 EventResponseStatusCode

An EventResponseStatusCode specifies the status codes that can be used by remote application receiving the event.

The following table provides the list of values that can be assigned to an EventResponseStatusCode.

EventResponseStatusCode	Description
200	The event was successfully received by the remote application.
404	The channelId in the event is unknown by the remote application. Cisco BroadWorks deletes the event channel and sends the event over the next event channel in the set.
481	The subscriptionId in the event notification is unknown to the remote application. Cisco BroadWorks deletes the subscription and drops the event.
500	The remote application encountered an unexpected condition that prevents it from handling the event. Cisco BroadWorks sends the event to the next event channel in the set.
400	The remote application could not process this specific event. Cisco BroadWorks drops the event.

8.4.5 ExternalApplicationId

An ExternalApplicationId identifies a remote application. It is provided by the remote application. An ExternalApplicationId is represented with a string.

8.5 Common Types

8.5.1 Address

An address maintains information about a SIP URI, tel URI, or digit string. The following type definition table lists the parameters of an address:

Parameter	Parameter Type	Required	Comments
address Info	String	Yes	The Address type contains the address information as either a SIP or tel URI or as a digit string.

Parameter	Parameter Type	Required	Comments
<i>countryCode</i>	CountryCode	No	The country code for the address. Only present if the address is an E.164 phone number. Note that this attribute is only for outgoing messages (messages sent by the Application Server). If present for an incoming message (message received by the Application Server), then it is ignored.

8.5.1.1 AlternateAddressType

An AlternateAddressType maintains information about one of the alternate addresses used by a subscriber. An AlternateAddressType can take one of the following possible values:

- An Integer from 0 through 10 identifying the alternate address.
- The string “Fax” when the alternate address is associated to a fax number.

An AlternateAddressType is represented with a string.

8.5.2 AccessEndpoint

An AccessEndpoint is an addressable location associated with a Subscriber. The following type definition table lists the parameters of an AccessEndpoint.

Parameter	Parameter Type	Required	Comments
<i>addressOfRecord</i>	string	Yes	In standalone mode, the address of record identifies a Line/Port for a device profile. In XS mode, the address of record identifies a public identity for an identity profile.

8.5.3 PartyInformation

A PartyInformation maintains information about a party. The following type definition table lists the parameters of a PartyInformation.

Parameter	Parameter Type	Required	Comments
<i>name</i>	string	No	The name of the party. Not present when there is no name or the user/application is not allowed to see it (the appropriate Calling Line Identification Presentation (CLIP)/ Connected Line Identification Presentation (COLP) service is not enabled, or privacy is restricted and the application is not trusted).
<i>address</i>	Address	No	The address of the party. Not present when there is no address (that is, the appropriate CLIP/COLP service is not enabled) or the user/application is not allowed to see it (that is, the privacy is restricted and the application is not trusted, see the privacy parameter).
<i>privacy</i>	EmptyContent	No	When present, indicates that CLIP/COLP presentation is restricted. If present, then the address parameter is not present. If not present, then the address parameter may be present (if available).

Parameter	Parameter Type	Required	Comments
<i>userId</i>	UserId	No	The user ID of the party. Only present when the party is in the same Group/Enterprise as the local user. Not present when the user/application is not allowed to see it (the appropriate CLIP/COLP service is not enabled, or privacy is restricted and the application is not trusted).
<i>callType</i>	String	No	The call type (that is, relationship) between the local user and this party. For more information, see section 8.5.3.1 CallType .
<i>userDN</i>	Address	No	The E164 address of the party. Only present when the party is in the same Group/Enterprise as the local user. Not present when there is no address or the user/application is not allowed to see it (the appropriate CLIP/COLP service is not enabled, or privacy is restricted and the application is not trusted).

8.5.3.1 CallType

A callType specifies the type of call associated with a party. The following table provides the list of values that can be assigned to a callType.

Call Type Value	Description	Call Session Topology
Group	The call session is between parties belonging to the same Cisco BroadWorks group.	Intra group/enterprise See section 4.1.9.1 Intragroup/Enterprise Topology .
Enterprise	The call session is between parties belonging to the same Cisco BroadWorks Enterprise.	Intra group/enterprise See section 4.1.9.1 Intragroup/Enterprise Topology .
Network	One of the parties connected to the call session is not a Cisco BroadWorks subscriber.	Extra-Cisco BroadWorks and Intra-Cisco BroadWorks topologies See sections 4.1.9.3 Extra-Cisco BroadWorks Topology and 4.1.9.2 Intra-Cisco BroadWorks Topology .
Emergency	The call session is connected to an emergency party (for example, 911).	Extra-BroadWorks topology See section 4.1.9.3 Extra-Cisco BroadWorks Topology .
Repair	The call session is connected to a repair party.	Extra-BroadWorks topology See section 4.1.9.3 Extra-Cisco BroadWorks Topology .
Private Dial Plan	The call session was established by using the Private Dial Plan configured in BroadWorks.	Extra-BroadWorks topology See section 4.1.9.3 Extra-Cisco BroadWorks Topology .
City-Wide Centrex	The call session was established by using the Private Dial Plan configured in BroadWorks for a City-Wide Centrex subscriber.	Extra-BroadWorks topology See section 4.1.9.3 Extra-Cisco BroadWorks Topology .
Unknown	The call type is unknown.	Extra-BroadWorks and Single call topologies See sections 4.1.9.3 Extra-Cisco BroadWorks Topology and 4.1.9.4 Single Call Topology .

Call Type Value	Description	Call Session Topology
Error	There was an error in determining the call type.	Extra-BroadWorks topology See section 4.1.9.3 Extra-Cisco BroadWorks Topology .

8.5.4 UserId

A UserId identifies a user or administrator. A UserId is in the form "user@domain". Note that the domain must be included. A UserId in the form "user" is not valid. A UserId is represented with a string.

8.6 Call Related Types

8.6.1 HookStatus

A HookStatus indicates the hook status of a subscriber. The following table provides the list of values that can be assigned to a HookStatus.

ConferenceType	Description
Off-Hook	Indicates that user is off-hook.
On-Hook	Indicates that user is on-hook.

8.6.2 Call

A call is an advanced type used by Cisco BroadWorks to describe the details of a call. A call is used in various events when call detail information is required. The following type definition table lists the parameters for a call.

Parameter	Parameter Type	Required	Comments
callId	CallId	Yes	The call ID of the call.
extTrackingId	ExternalTrackingId	Yes	The external tracking ID for a call.
networkCallId	String	No	This parameter is only present when the remote party is a network party. For a call with originator personality, this parameter is set to the SIP Call ID of the outgoing SIP INVITE. For a call with terminator personality, it is the SIP Call ID of the incoming SIP INVITE.
personality	Personality	Yes	The personality of a call.
state	CallState	Yes	The state of a call.
releaseCause	ReleaseCause	No	The release cause for the call. Only present if the release was due to an explicit release of the call due to an error. For more information, see section 8 Type Definitions .
remoteParty	PartyInformation	Yes	Information about the remote party. For more information, see section 8 Type Definitions .
redirections	redirectionList	No	List of all the redirections (if any) for the call. These are the details of other parties that have redirected this call. Only present for unanswered, incoming calls. For more information, see section 8 Type Definitions .



Parameter	Parameter Type	Required	Comments
<i>redirect</i>	Redirect	No	The details for the redirect when the local user has redirected the call. For information, see section 8 Type Definitions .
<i>endpoint</i>	AccessEndpoint	No	The details of the endpoint that is linked to the call. Only present when an endpoint is associated with the call. For more information, see section 4.1.7 Endpoint .
<i>appearance</i>	Appearance	No	The appearance value for the call. Only present when the call has an appearance value assigned. This value matches the appearance index in the context of Enhanced Shared Call Appearance.
<i>allowAnswer</i>	EmptyContent	No	When present, an application can answer a call in the Alerting state with a Terminator personality or Click-to-Dial personality using a Talk request. When not present, an application cannot answer a call in the Alerting state with a Terminator personality or Click-to-Dial personality using a Talk request. Note that this parameter only applies to calls in the Alerting state with Terminator personality and Alerting state with Click-to-Dial personality.
<i>allowRetrieve</i>	EmptyContent	No	When present, an application can retrieve a held call using a Talk request. When not present, an application cannot retrieve a held call.
<i>alternateAddressType</i>	AlternateAddressType	No	The alternate address type in use for the local user. Only present when an alternate address is in use. It is not present when the user's primary address is in use.
<i>distinctiveRing</i>	DistinctiveRing	No	The distinctive ring pattern in use for the call. Only present for incoming calls for which a distinctive ring pattern should apply. For more information, see section 8.6.2.13 DistinctiveRing .
<i>diversionInhibited</i>	EmptyContent	No	When present, indicates that diversions are inhibited for the call and diversion/redirection attempts (including those initiated by applications) may be rejected. Redirection attempts from an application are rejected if the call is an Alerting, Terminator call, but are allowed for other types of calls.
<i>recallType</i>	RecallType	No	The recall type for the call. Only present if the call was triggered by a callback or recall tracked by RecallType.
<i>startTime</i>	Timestamp	Yes	The start time for the call. The value matches the value in the CDR for the call. Note that in some scenarios (for example, Click-to-Dial call or a Barge-in); the start time for the call can change.
<i>answerTime</i>	Timestamp	No	The answer time for the call. Only present if the call was answered. The value matches the value in the CDR for the call. Note that in some scenarios (Directed Call Pickup with Barge-in, Monitor Call, and Emergency Call to Supervisor); the answer time for the call can change (as explained in section 5 Requests).



Parameter	Parameter Type	Required	Comments
<i>releaseTime</i>	Timestamp	No	The release time for the call. Only present when the call is reported as Released. For notifications where a Detached call is reported as Released, the value is actually the time at which the call was Detached. For notifications where the call is truly being released and not Detached, the value matches the value in the CDR for the call.
<i>detachedTime</i>	Timestamp	No	The detached time for the call. Only present when the call entered the <i>Release</i> state.
<i>releasingParty</i>	ReleasingParty	No	Specify if the call was released locally or by the remote party. Only present when the call is released.
<i>heldTime</i>	Timestamp	No	The time this call entered the <i>Held</i> state. Present only if the call is in the <i>Held</i> state.
<i>totalHeldTime</i>	PositiveInteger	No	The total cumulated time the call spent in the <i>Held</i> state in milliseconds. Only present if the call was at least once in the <i>Held</i> state.
<i>agentEscalation (AS only)</i>	AgentEscalation	No	Specify the agent escalation type. Only present when the call is in relation to an agent escalation.
<i>acdCallInfo (AS only)</i>	ACDCallInfo	No	Contains the details of an ACD call. For more information, see section 8.6.2.12 ACDCallInfo .
<i>huntGroup UserId (AS only)</i>	UserId	No	The user ID of the hunt group being used to reach the agent (the local user). Only present when the call is an unanswered hunt group termination.
<i>acdOutgoing CallInfo (AS only)</i>	ACDOutgoing CallInfo	No	The ACD call information when an agent makes an outgoing call as an ACD.
<i>groupCallParkLocationSelected (AS only)</i>	EmptyContent	No	When present, indicates that the Group Call Park feature has been invoked and the parked location selected. This indicates that the <i>remoteParty</i> element contains the Parked location.
<i>recorded (AS only)</i>	EmptyContent	No	When present, indicates that the call is being recorded.
<i>findMeFollowMeUserId (AS only)</i>	UserId	No	The user ID of the Find-Me/Follow-Me user being used to redirect the call. It is present when the call is being redirected to the subscriber by a Find-Me/Follow-Me group of which they are a member.
<i>allowedRecordingControls (AS only)</i>	RecordingControls	No	The recording controls that are allowed for this call. Present when the user has the Call Recording service active.
<i>recordingState (AS only)</i>	RecordingState	No	The current state of the recording. Present when the user has the Call Recording service active and a call recording is currently pending, started, paused, or failed. Absent when there is no active or pending recording.
<i>executiveUserId (AS only)</i>	UserId	No	The user ID of the Executive for the Executive Assistant service. Present when the call is being initiated by or filtered for an executive assistant for the Executive Assistant Service.
<i>recallFromParty (AS only)</i>	PartyInformation	No	The recall From Party for a Recall action. Present for recalls from Consultative Call Transfer, Blind Transfer, and Call Park.

Parameter	Parameter Type	Required	Comments
<i>securityClassification (AS only)</i>	NonEmptyToken	No	The security classification level of the call. Present when the Security Classification service is assigned to the user and the call classification level has been determined.
<i>clientSessionInfo (AS only)</i>	ClientSessionInfo	No	An opaque token set by the client. Present if client session info has been set for this session.

The following subsections provide additional information regarding the type used by the Call type.

8.6.2.1 CallId

A callId identifies a call. A callId is of the form “localhost-<parameter1>:<parameter2>”, where parameter1 and parameter2 are *NonNegativeIntegers*. They are incremented based on internal Cisco BroadWorks logic. When one of them reaches the *NonNegativeInteger* maximum value ($2^{31}-1$), it rolls back to 0.

The value “localhost286:0” is an example of a callId used by Cisco BroadWorks.

A CallId is represented with a string.

8.6.2.2 ExternalTrackingId

An ExternalTrackingId uniquely identifies a call session. An external tracking ID is of the form “<parameter1>:<parameter2>” where:

Parameter	Description
parameter1	NonNegativeInteger that is incremented each time a call session is created on a Cisco BroadWorks server. When it reaches the NonNegativeInteger maximum value ($2^{31}-1$) it rolls back to 0.
parameter2	Cisco BroadWorks server identifier on which the call was created. This identifier is unique within an Application Server cluster.

The value “6:1” is an example of an ExternalTrackingId.

An ExternalTrackingId is represented with a string.

8.6.2.3 Personality

A Personality maintains information about a call personality. The following table provides the list of values that can be assigned to a Personality.

Personality	Description
Originator	Personality of the call when the subscriber is the initiator of the call session. If A initiates a call session to destination B, then the call associated to A has its personality set to “Originator”.
Terminator	Personality of the call when the subscriber receives an incoming call. If A initiates a call to destination B, then the call associated to B has its personality set to “Terminator”
Click-to-Dial	Personality of the call when a “dial” request is being sent on behalf of a subscriber. If a remote application tries to establish a call session from A to B (through a third-party call control dial request) then the call associated to A has its personality set to “Click-To-Dial” until A answers. Once A has answered, then a call attempt is made by Cisco BroadWorks to B and A’s personality is set to “Originator”.

8.6.2.4 CallState

A CallState maintains information about the state of a call. The following table provides the list of values that can be assigned to a CallState.

CallState	Description
Alerting	Corresponds to the <i>Alerting</i> state.
Active	Corresponds to the <i>Active</i> state.
Held	Corresponds to the <i>Held</i> state.
Remote Held	Corresponds to the <i>Remote Held</i> state.
Detached	Corresponds to the <i>Detached</i> state.
Released	Corresponds to the <i>Released</i> state.
Idle	Corresponds to the <i>Idle</i> state.

8.6.2.5 Appearance

An Appearance matches the appearance index in the context of Enhanced Shared Call Appearance. An Appearance is represented with a PositiveInteger.

8.6.2.6 Timestamp

A Timestamp represents the number of milliseconds elapsed since January 1, 1970 UTC. A Timestamp type is maintained in a Long type.

8.6.2.7 ReleaseCause

A ReleaseCause maintains information about the release of a call. A ReleaseCause is only set if the call is being released due to a failure of some kind. Note that it can be present for calls that have not yet been released such as calls receiving treatment and Detached calls.

The following type definition table lists the parameters of a ReleaseCause.

Parameter	Parameter Type	Required	Comments
<i>internalReleaseCause</i>	InternalReleaseCause	Yes	The reason the call was released as used by Cisco BroadWorks internally. For more information, see section 8 Type Definitions .
<i>q850Cause</i>	String	No	This value represents the Integrated Services Digital Network (ISDN) cause code. This value is also sent inside the SIP Reason header if configured to do so.
<i>SIPStatus</i>	String	No	This is the status code and message used in the SIP response. This status is also included in the SIP entry of the Reason header for CANCEL and BYE requests.
<i>cdrTerminationCause</i>	CdrTerminationCause	Yes	The call detail record terminationCause is defined as a three-digit string that indicates the reason why the call completed. For more information, see section 8 Type Definitions .

8.6.2.7.1 InternalReleaseCause

An InternalReleaseCause specifies the reason the call was released by Cisco BroadWorks internally. The following table provides the list of values that can be assigned to an internalReleaseCause.

Reason	Description
Busy	As its name implies, this cause is used when the call cannot complete because the resources are currently in use for another call. Note that this Release Cause is also used by the Do Not Disturb service. This cause triggers the Call Forwarding Busy service and not the Call Forwarding No Answer service. It also triggers transfer to Voice Mail, but with specific Busy conditions.
Forbidden	This Release Cause is currently used by Call Blocking services and Call Processing to indicate that the call cannot complete because the current system configuration does not allow it. This cause is also used for some specific SIP status codes and when call loops are detected. Note that services that react on call releases do not trigger on FORBIDDEN Release Causes if they are received from services (for example, the call is not transferred to Voice Mail when blocked by Incoming Calling Plan [ICP]). These services include Voice Mail (VM), Call Forwarding No Answer (CFNA), and Call Forwarding Not Reachable (CFNR).
Routing Failure	When Cisco BroadWorks cannot route a call because no route list was built, it uses this Release Cause.
Global Failure	This Release Cause is used as a default value for 6xx SIP Response Codes. It is not generated internally and does not trigger any special service behavior.
Request Failure	This is the default, non-specific Release Cause for use with 4xx SIP responses. This Release Cause triggers route advancing in the network router.
Server Failure	This is the Release Cause used for all 5xx SIP responses. It also triggers route advancing in the network router.
Translation Failure	When Cisco BroadWorks encounters invalid dialed digits, it uses the TRANSLATION_FAILURE Release Cause to release the call. This Release Cause does not trigger any specific service behavior.
Temporarily Unavailable	This Release Cause is used in any situation that is deemed temporary (that is, the call should work the next time it is tried, or in the near future). For example, communication problems with the Media Server yields a TEMPORARILY_UNAVAILABLE release, as this is most likely due to a transitional fault. Another typical use of this Release Cause is when a call is placed to an unregistered device.
User Not Found	As the name implies, this Release Cause is used when a call terminates to Cisco BroadWorks, but the user cannot be found. This differs from TRANSLATION_FAILURE in that the dialed digits are valid, but do not correspond to a user configured within Cisco BroadWorks.
Request Timeout	Again, as the name indicates, this Release Cause is used when the request has not been answered in a short enough time. The timeout depends on the request and situation. This Release Cause triggers route advancing in the network router.
Dial Tone Timeout	This Release Cause is used when the caller did not enter any digits before the timeout. It does not trigger any special service behavior.

Reason	Description
Insufficient Credits	<p>This Release Cause is used when the Prepaid service releases a call because the user runs out of credits. Note that this Release Cause is used locally for the Prepaid user, but not for the remote party who is oblivious to this situation.</p> <p>Note that services that react on call releases do not trigger on NO_MORE_PREPAID_CREDITS Release Causes (for example, the call is not transferred to Voice Mail if it is not granted credits). These services include Voice Mail (VM), Call Forwarding No Answer (CFNA), and Call Forwarding Not Reachable (CFNR).</p>

For more information about the *InternalReleaseCause* values, see the *Cisco BroadWorks Treatment Guide* [1].

8.6.2.7.2 TerminationCause

A CdrTerminationCause specifies the completion reason by the accounting interface. A CdrTerminationCause is represented with a string.

8.6.2.8 RedirectionList

A Redirection contains a list of Redirections.

8.6.2.9 Redirection

A Redirection provides information about the redirecting party. The following type definition table lists the parameters of a Redirection.

Parameter	Parameter Type	Required	Comments
<i>party</i>	PartyInformation	No	The details of the redirecting party. For more information, see <i>PartyInformation</i> .
<i>redirectReason</i>	RedirectReason	No	The reason for the redirection.
<i>counter</i>	PositiveInteger	No	The number of redirections represented by this redirect entry. This is the same as the counter parameter in a SIP Diversion header entry. If the counter parameter is not present, it is the same as if the counter parameter was present and set to 1.

8.6.2.10 RedirectionReason

A RedirectReason specifies the reason for a redirection. The following table provides the list of values that can be assigned to a RedirectReason.

Reason	Description
user-busy	The call session was redirected to a specified destination because Call Forward Busy service is activated and the subscriber is busy. A subscriber is considered busy when there are too many active calls or when the Do Not Disturb service is activated.
no-answer	The call session was redirected because Call Forward No Answer service was activated and the subscriber did not answer within the specified number of rings.
unavailable	The call session was redirected because Call Forwarding Not Reachable service was activated and the subscriber device is unreachable or unregistered.
unconditional	The call session was redirected because the Call Forwarding Always (CFA) service was activated. When active, CFA causes all incoming calls to be redirected unconditionally.

Reason	Description
time-of-day	The call session was redirected because Call Forwarding Selective (CFS) was activated and the incoming call matched the pre-specified time-of-day criteria.
do-not-disturb	The call session was redirected by a 3rd party Do Not Disturb service prior to reaching Cisco BroadWorks.
deflection	Used by the Redirect type when call is being redirected for other reasons than a transfer. Used by the Redirection type when the call has been transferred.
follow-me	The call session was redirected because one of the following services was activated: Simultaneous Ringing, Sequential Ringing, BroadWorks Anywhere, or Remote Office.
out-of-service	The call session was redirected by the network with the reason set to "out-of-service".
away	The call session was redirected by the network with the reason set to "away".
transfer	Used by the Redirect type. Indicates that the call is being redirected following a transfer.
Voice mail	The call session was redirected because Call Forward Always to Voice Mail was activated, or the Call Forward Busy to Voice Mail was activated and the subscriber was busy, or the Call Forwarding No Answer to Voice Mail was activated and there was no answer.
hunt-group	The call session was redirected by a hunt group to an agent.
call-center	The call session was redirected by a call center to an agent.
unknown	The call session was redirected for unknown reason.

8.6.2.11 Redirect

A Redirect provides information about a call redirection. The following type definition table lists the parameters of a Redirect.

Parameter	Parameter Type	Required	Comments
<i>address</i>	Address	No	The target address for the redirect. Usually present, however, it is not present in certain special scenarios; for example, transfer with consultation, where the transferring party does not have CLIP enabled or privacy is restricted for the transferred party.
<i>redirectTime</i>	Timestamp	Yes	The time at which the redirect took place.
<i>reason</i>	RedirectReason	Yes	The reason for the redirection.

8.6.2.12 ACDCallInfo

An ACDCallInfo is provided when a call is delivered to an agent by an ACD or a Route Point. The following type definition table lists the parameters of an ACDCallInfo.

Parameter	Parameter Type	Required	Comments
<i>acdUserId</i>	UserId	Yes	The user ID of the ACD being used to reach the agent (the local user).
<i>acdName</i>	String	Yes	The name configured against the ACD.
<i>acdNumber</i>	Address	Yes	The address of the ACD being used to reach the agent. This could be the primary DN or a DNIS configured against the ACD.

Parameter	Parameter Type	Required	Comments
<i>numCallsInQueue</i>	NonNegativeInt	Yes	The number of calls remaining in the queue.
<i>waitTime</i>	NonNegativeInt	Yes	The amount of time (in seconds) that the caller has spent in a queue waiting for an agent.
<i>longestWaitTime</i>	NonNegativeInt	No	The longest wait time (in seconds) of the calls remaining in the queue.
<i>HoldReminder</i>	EmptyContent	No	When present, indicates that an ACD call has been held for longer than the configured time threshold. A client application may use this information to display highly visible indicator to remind the agent to retrieve the call. It also indicates that the call may bounce back to the queue if not retrieved before the configured time threshold.
<i>callingPartyInfo</i>	PartyInfo	Yes	The information of the calling party. Contains the real information for the calling party, unlike the call type's remoteParty element, which may be modified for call center display purposes, such as by prepending the call center name to the party's name or replacing the party's address with a DNIS address for the call center.
<i>routePoint</i>	EmptyContent	No	Indicates whether the incoming call is being offered from a route point. This field may be used to distinguish the origin of the incoming call, that is, call center or route point.
<i>DispositionCode</i>	AgentDispositionCode	No	When present, indicates the default disposition code applicable for this call.
<i>postCallAgentState</i>	String	No	When present indicates that the option to "Automatically set agent state to "x" after call" is enabled. The string value indicates the agent state after call.
<i>postCallAgentState</i>	String	No	When present indicates that the option to "Automatically set agent state to "x" after call" is enabled. The string value indicates the agent state after call.

8.6.2.13 DistinctiveRing

A DistinctiveRing specifies a ringing pattern. The following table provides the list of values that can be assigned to a DistinctiveRing.

DistinctiveRing	Description
Bellcore-dr2	Distinctive ringing pattern 2 as defined in GR-506-CORE. This is used by the Priority Alert service, and for the Long-Long ring pattern of the Alternate Numbers service.
Bellcore-dr3	Distinctive ringing pattern 3 as defined in GR-506-CORE. This is used for the Short-Short-Long ring pattern of the Alternate Numbers service.
Bellcore-dr4	Distinctive ringing pattern 4 as defined in GR-506-CORE. This is used for the Short-Long-Short ring pattern of the Alternate Numbers service.

8.6.2.14 ReleasingParty

A ReleasingParty identifies the party that releases the call. The following table provides the list of values that can be assigned to a ReleasingParty.

ReleasingParty	Description
localRelease	Specify that the release was performed by the subscriber associated with the call.
remoteRelease	Specify that the release was performed by the remote party.

8.6.2.15 RecallType

A RecallType identifies which service triggers a callback. The following table provides the list of values that can be assigned to a RecallType.

Reason	Description
Busy Camp On Callback	The callback was performed because the camp-on call remained unanswered beyond the configurable period.
Call Transfer Recall	The callback was performed because the transferred call failed to reach the destination.
Call Park Recall	The callback was performed because the parked call was not retrieved within the configurable period.
Automatic Hold/Retrieve Recall	The callback was performed because the held call was not retrieved within the configurable period.
Executive-Assistant Call Push Recall	The callback was performed because the call initiated by the executive assistant was not answered by the executive.

8.6.2.16 AgentEscalation

An AgentEscalation specifies the agent escalation call type. The following table provides the list of values that can be assigned to an AgentEscalation.

AgentEscalation	Description
Normal	Specify that the call escalation level is normal (following an Escalate To Supervisor request).
Emergency	Specify that the call escalation level is set to "emergency" (following an Emergency Call to Supervisor request).

8.6.2.17 ACDOutgoingCallInfo

An ACDOutgoingCallInfo contains the details of an outgoing call made by an agent as an ACD. The following type definition table lists the parameters of an ACDOutgoingCallInfo.

Parameter	Parameter Type	Required	Comments
acdUserId	UserId	Yes	The user ID of the ACD that the agent is making the call as.
acdName	String	Yes	The name configured against the ACD.
acdNumber	Address	Yes	The address of the ACD being used for the outgoing call. This could be the primary DN or a DNIS configured against the ACD.

Parameter	Parameter Type	Required	Comments
<i>routePoint</i>	EmptyContent	No	Indicates whether the incoming call is being offered from a route point. This field may be used to distinguish the origin of the incoming call, that is, call center or route point.
<i>dispositionCode</i>	AgentDispositionCode	No	When present, indicates the default disposition code applicable for this call.

8.6.2.18 RecordingControls

A RecordingControls list the currently allowed recording controls. The following table provides the list of values that can be assigned to a RecordingControls.

AgentEscalation	Description
none	No recording controls are valid. This is sent when the user has the <i>Always</i> recording mode.
record	The user can keep the recording or start recording the call. This is sent if a user has the <i>On Demand</i> or <i>On Demand with User Initiated Start</i> mode.
pause	The user can pause the recording. This is sent when a call is recording and the user's recording mode is <i>On Demand</i> once the user has kept the recording, or this is sent when the user has the <i>Always with Pause/Resume</i> mode.
Pause-stop	The user can pause or stop the recording. This is sent once a user with the <i>On Demand with User Initiated Start</i> mode has started recording a call.
Resume	The user can resume a paused recording. This is sent once a user has successfully paused a recording and has either the <i>On Demand</i> or the <i>Always with Pause/Resume</i> mode.
Resume-stop	The user can resume or stop a recording. This is sent once a user with the <i>On Demand with User Initiated Start</i> mode has paused the recording of a call.
stop	The user can stop the recording. If the user's recording mode allows for it, this control is sent when the recording state is <i>pending</i> . This gives the user the option to cancel the pending recording of the held call. No recording is started as long as the user does not take the call off hold while requesting the stop of the pending recording.

8.6.2.19 RecordingState

A RecordingState specifies the state of a recording. The following table provides the list of values that can be assigned to a RecordingState.

AgentEscalation	Description
Pending	Call recording has been requested but has not yet been started. This can happen when the recording request is received while a call is held.
Started	The call is being recorded.
Paused	Call recording was started and is now paused.
Failed	Call recording was started but failed. In this state, a new call recording cannot be started.

8.6.2.20 ClientSessionInfo

A ClientSessionInfo is an opaque token used by a client to correlate sessions.

A ClientSessionInfo is represented with a string.

8.6.3 CallInfoList

A CallInfoList maintains a list of CallInfo parameters.

8.6.4 CallInfo

A CallInfo maintains summary information about a call. The following type definition table lists the parameters of a CallInfo.

Parameter	Parameter Type	Required	Comments
<i>callId</i>	CallId	Yes	The call ID of the call.
<i>Uri</i>	String	No	This parameter is not used by CTI interface. Only used when running over HTTP.
<i>inConference</i>	EmptyContent	No	Only present if the call is in a conference.

8.6.5 CallIdList

A CallIdList maintains a list of sorted call IDs.

8.6.6 CallList

A CallIdList maintains a list of call parameters, sorted by call ID.

8.6.7 CallRecordingMode

A *CallRecordingMode* specifies a call recording mode. The following table provides the list of values that can be assigned to a *CallRecordingMode*.

CallRecordingMode	Description
<i>always</i>	This mode indicates that all calls originated by or received by the user are automatically recorded and saved to the designated recording platform.
<i>never</i>	This mode indicates that calls are never recorded.
<i>on-demand</i>	This mode indicates that calls can be selectively recorded and saved by the user on an on-demand basis.
<i>always-pause-resume</i>	This mode indicates that calls originated or received by the user are always recorded and saved. The user also has the ability to pause and resume recordings.
<i>on-demand-user-start</i>	This mode indicates that the call recording does not start until initiated by the user. The user also has the ability to pause, resume, and stop recordings.

8.6.8 CallStartInfo

A CallStartInfo identifies a new created call. The following type definition table lists the parameters of a CallStartInfo.

Parameter	Parameter Type	Required	Comments
<i>callId</i>	CallId	Yes	The call ID of the created call.
<i>externalTrackingId</i>	ExternalTrackingId	Yes	The external tracking ID of the created call.

8.6.9 CallMeNowStartInfo

A CallMeNowStartInfo provides the passcode that the external party must dial upon answering the incoming call that results from the Call Me Now request.

Parameter	Parameter Type	Required	Comments
<i>Passcode</i>	CallMeNowPasscode	No	The passcode to use for the Call Me Now call. Only present when the Call Me Now service's answer confirmation option set to Passcode.

8.6.10 CallMeNowPasscode

The passcode used by the Call Me Now service. It is composed of digits (0-9) and is 2 to 30 digits long per system configuration.

8.6.11 ClickToDialLocationEnum

A ClickToDialLocationEnum specifies location type to alert for Click-to-Dial type origination. The following table provides the list of values that can be assigned to a ClickToDialLocationEnum.

ClickToDialLocationEnum	Description
Primary	This indicates that the primary location should be alerted.
BroadWorksAnywhere	This indicates that BroadWorks Anywhere locations or a specified BroadWorks Anywhere location should be alerted.
SharedCallAppearance	This indicates that Shared Call Appearance locations or a specified Shared Call Appearance location should be alerted.
Mobility	This indicates that BroadWorks Mobility locations should be alerted.
FlexibleSeatingGuest	This indicates that the Flexible Seating Guest location should be alerted.
All	This indicates that the primary location, the BroadWorks Anywhere locations, the Shared Call Appearance locations, and the BroadWorks Mobility locations should be alerted.

8.6.12 IMRNInfo

An IMRNInfo contains the allocated IMRN information for a mobile originated or terminated call. The following type definition table lists the parameters of an IMRNInfo.

Parameter	Parameter Type	Required	Comments
<i>imrn</i>	DN	Yes	The allocated IMRN in E164 format.

8.7 Conference Related Types

8.7.1 Conference

A Conference maintains information about a conference. The following table lists the parameters of a Conference.

Parameter	Parameter Type	Required	Comments
<i>state</i>	ConferenceState	Yes	The state of the conference.
<i>endPoint</i>	AccessEndPoint	No	The conference endpoint. Only present when an endpoint is associated with the conference.
<i>appearance</i>	Appearance	No	The appearance value for the conference. Only present when the conference has an appearance value assigned. This value matches the appearance index in the context of Enhanced Shared Call Appearance.
<i>conferenceType</i>	ConferenceType	No	When present, indicates a specific type of conference like Silently Monitored conference or Barge-In conference and so on.
<i>ConferenceParticipantList</i>	ConferenceParticipantList	No	The list of conference participants.
<i>muted</i>	EmptyContent	No	When present, indicates that the controller of the conference is muted (for example, as a result of a Monitor Call request).

8.7.1.1 ConferenceState

A ConferenceState maintains state information about a conference. A ConferenceState is represented with a string.

8.7.1.2 ConferenceType

A ConferenceType maintains state information about a conference type. The following table provides the list of values that can be assigned to a ConferenceType.

ConferenceType	Description
Barge-in	Specify that the conference was established as a result of a barge-in operation.
Silent Monitor	Specify that the conference was established as a result of a silent monitoring operation.

8.7.1.3 ConferenceParticipant

A ConferenceParticipant maintains information about a conference participant. The following type definition table lists the parameters of a ConferenceParticipant.

Parameter	Parameter Type	Required	Comments
<i>callId</i>	CallId	Yes	The call ID of the call.
<i>muted</i>	EmptyContent	No	When present, indicates that the call has been muted.

Parameter	Parameter Type	Required	Comments
<i>deaf</i>	EmptyContent	No	When present, indicates that the call is deaf (that is, not transmitting the audio stream to the participant).

8.7.1.4 ConferenceParticipantList

A ConferenceParticipantList contains a list of ConferenceParticipant.

8.8 Route Point Related Types

8.8.1 RoutePointQueue

A RoutePointQueue provides information about a route point queue. The following type definition table lists the parameters of a RoutePointQueue.

Parameter	Parameter Type	Required	Comments
<i>queueEntries</i>	RoutePointQueueEntryList	No	The list of queue entries maintained in the route point queue.

8.8.2 RoutePointQueueEntryList

A RoutePointQueueEntryList contains a list of RoutePointQueueEntry.

8.8.3 RoutePointQueueEntry

A RoutePointQueueEntry provides information about each call maintained in a route point queue. The following type definition table lists the parameters of a RoutePointQueueEntry.

Parameter	Parameter Type	Required	Comments
<i>callId</i>	CallId	Yes	The call ID of the call for the route point entry.
<i>extTrackingId</i>	ExternalTrackingId	Yes	The external tracking ID of the call for the route point entry.
<i>remoteParty</i>	PartyInformation	Yes	Information about the remote party for the route point entry.
<i>addTime</i>	Timestamp	No	The time when this entry was added to the route point queue. Only present if the call has been added to the queue.
<i>removeTime</i>	Timestamp	No	The time when this entry was removed from the route point queue. Only present when the entry has been removed from the queue.
<i>bounced</i>	EmptyContent	No	When present, indicates that the call was at some point offered to an agent and bounced.
<i>routePointNumber</i>	Address	Yes	Indicates the Route Point number the call was received on.
<i>routePointName</i>	String	Yes	Indicates the Route Point name associated with the RP number the call was received on.



Parameter	Parameter Type	Required	Comments
<i>playCollectInfo</i>	PlayCollectionInfo	No	Details on the play collection operation in progress. Only present if a play-collect is currently being performed.
<i>outgoingCall</i>	EmptyContent	No	When present, indicates that the call was added as a result of an outgoing Dial request. If no present, then the call was added as a result of incoming call.
<i>outgoingCallAnswerTime</i>	Timestamp	No	Specify when the queued call was answered by the callee. Only present for outgoing call that have been answered by the called destination.
<i>answeringUserId</i>	UserId	No	The user ID of the agent who answered the call. This parameter is only present when the agent who answered the call is a Cisco BroadWorks subscriber.
<i>answeringCallId</i>	CallId	No	The call ID of the agent's call (when the call is answered by an agent). This parameter is only present when the call offer from the call center to the agent does not contain a SIP call leg. This parameter can be used to correlate with events generated for a call.
<i>answeringNetworkCallId</i>	String	No	The SIP call ID for the call when the call is answered by an agent. This parameter is only present when the call offer from the call center to the agent contains a SIP call leg. This parameter can be used to correlate with events generated for a call.

8.8.4 RoutePointState

A RoutePointState maintains state information about a Route Point. The following table provides the list of values that can be assigned to a RoutePointState.

RoutePointState	Description
Normal	All call received by the Route Point are queued and treated normally.
Failed	All call received by the Route Point are sent to the failed destination.



8.8.5 StateChangeReason

A StateChangeReason specifies the reason for a Route Point state change. The following table provides the list of values that can be assigned to a StateChangeReason.

StateChangeReason	Description
FailureDetected	The state change was performed because a failure was detected.
ExternalRequest	The state change was performed because it was requested through an explicit request.

8.8.6 OverflowReason

An OverflowReason indicates the cause for the overflow. The following table provides the list of values that can be assigned to an OverflowReason.

OverflowReason	Description
Size	The call was overflowed because the queue had reached its maximum capacity when trying to add the call.
Time	The call was overflowed because the call was not handled within the configured amount of time.

8.8.7 RoutePointCallFailureReason

A RoutePointCallFailureReason specifies the cause for a Route Point call failure. The following table provides the list of values that can be assigned to a RoutePointCallFailureReason.

RoutePointCallFailureReason	Description
CallFailureRequested	The call was marked as failed as requested by the remote application through a Route Point Fail Call.
CallFailureDueToInactivity	The call was marked as failed because no request was received for the call within the configurable time period.
RoutePointFailure	The call was marked as failed because the Route Point is in the <i>Failed</i> state.

8.8.8 RoutePointCallReleaseReason

A RoutePointCallReleaseReason specifies the cause for a Route Point call release. The following table provides the list of values that can be assigned to a RoutePointCallReleaseReason.

RoutePointCallReleaseReason	Description
CallReleaseRequested	The call was released as requested by the remote application through a Route Point Release Call.
NoAnswerTimeOut	The call was released because the no answer timer expired prior to the call being answered. This reason is only applicable for outgoing calls.
ReleasedWithCause	The call was released because the call failed to be delivered to the provided destination. This reason is only applicable for outgoing calls.

8.8.9 CPDResult

A CPDResult provides information regarding the Call Progress Detection performed on an outgoing call. The following table provides the list of values that can be assigned to a CPDResult.

CPDResult	Description
SIT	Special Information Tone was received during call setup.
Fax	A Fax machine answered the call.
VoiceMessaging	A voice messaging system answered the call.

8.8.10 NonEmptyURI

A URI is a Uniform Resource Identifier. A URI is represented with a string.

8.8.11 RoutePointCallStartInfo

A RoutePointCallStartInfo identifies a new created call. The following type definition table lists the parameters of a CallStartInfo.

Parameter	Parameter Type	Required	Comments
<i>callId</i>	CallId	Yes	The call ID of the created call.
<i>externalTrackingId</i>	ExternalTrackingId	Yes	The external tracking ID of the created call.

8.8.12 Media Playback and Digit Collection Types

8.8.12.1 PlayCollectInfo

A PlayCollectInfo provides information about a play-collect operation. The following type definition table lists the content of a PlayCollectInfo.

Parameter	Parameter Type	Required	Comments
<i>playCollectId</i>	PlayCollectId	Yes	PlayCollectId as returned in the response. Use to identify the operation and match the request with its associated events.
<i>playCollectType</i>	PlayCollectType	Yes	The type of play collection operation.

8.8.12.1.1 *PlayCollectId*

A playCollectId identifies a playCollect operation. A playCollectId is represented with a string.

8.8.12.1.2 *PlayCollectType*

A playCollectType specifies the nature of the play-collect operation performed. The following table provides the list of values that can be assigned to a playCollectType.

playCollectType	Description
Treatment	Indicates that a treatment is being played.
MOH	Indicates that Music On Hold is being played.
Ringback	Indicates that a ringback is being played.

playCollectType	Description
Busy	Indicates that a busy is being played
Silence	Indicates that silence is being played.

8.8.12.2 PlayCollectResult

A PlayCollectResult provides information regarding the completion of a play-collect operation. The following type definition table lists the content of a PlayCollectResult.

Parameter	Parameter Type	Required	Comments
<i>playCollectId</i>	PlayCollectId	Yes	PlayCollectId as returned in the response. Use to match the request with its associated events.
<i>reason</i>	CompletionReason	Yes	The reason the operation was completed.
<i>digits</i>	String	No	Contains the collected DTMF input characters. Only present if a DTMF input was collected.
<i>errorReason</i>	String	No	Contains detail of the error. Only present when the reason is set to "Error".

8.8.12.2.1 CompletionReason

A CompletionReason indicates the reason the play-collect operation has ended. The following table provides the list of values that can be assigned to a CompletionReason.

CompletionReason	Description
Match	Indicates that the digit map was matched.
NoMatch	Indicates that the digit map was not matched. Can only be returned when collecting more than one digit sequence.
Timeout	Indicates that no DTMF input was received before one of the collection timers expired.
EOF	Indicates that the play operation has completed as expected. This reason is only used when no digits are collected.
Stopped-OutOfQueue	Indicates that the play-collect operation was implicitly stopped because the call was removed from the Route Point queue.
Stopped-NewPlayCollect	Indicates that the play-collect operation was stopped because a new play-collect was requested for the call.
Stopped-OfferedWithRingback	Indicated that the play-collect operation was implicitly stopped because the call was offered to an agent and the system is now providing ringback as per the Route Point configuration.
Error	Indicates that the play or DTMF collection operation was not completed because off an error. The error reason is provided in the <i>errorDetail</i> parameter of the PlayCollectResult.

8.8.12.3 BusyPattern

A BusyPattern specifies a busy signal. The following table provides the list of values that can be assigned to a BusyPattern.

BusyType	Description
Normal	Corresponds to the localized busy signal.
FastBusy	Corresponds to the localized fast busy signal.

8.8.12.4 CallUpdateReason

A CallUpdateReason specifies the reason a *Route Point Call Updated* event or an *ACD Call Updated* event was generated. The following table provides the list of values that can be assigned to a CallUpdateReason.

BusyType	Description
Call Transferred	The call was transferred by the remote party.
Call Picked-up	The call was picked-up by the remote party as a result of a Directed Call Pickup request or Directed Call Pickup with Barge-in where the call state was in <i>Alerting</i> .
Call Barged-In	The call was barged-in as a result of a Directed Call Pickup with Barge-in request.
Call Monitored	The call is being monitored as a result of a Monitor Call or Monitor Next Call request.
Call Forwarded	The call is being forwarded as a result of the application of a Cisco BroadWorks service (for example, Call Forward Always).

8.8.12.5 RoutePointPlayTreatmentURIList

A RoutePointCallFailureReason contains a list of up to four NonEmptyURIs that are used in the context of a RoutePointPlayTreatmentRequest.

8.8.12.6 RoutePointPlayTreatmentNumberOfPlay

A RoutePointPlayTreatmentNumberOfPlay indicates the number of times that the treatment is to be played. A RoutePointPlayTreatmentNumberOfPlay is represented with a PositiveInteger smaller than 5.

8.8.12.7 RoutePointPlayInfo

A RoutePointPlayInfo carries the playCollectId to correlate the request with its associated event when a Play Request (PlayBusy, PlaySilence, and so on.) is made successfully. The following type definition table lists the content of a RoutePointPlayInfo.

Parameter	Parameter Type	Required	Comments
playCollectId	PlayCollectId	Yes	The playCollectId used to correlate the request with its associated event.

8.8.13 RoutePointDistribute

A RoutePointDistribute provides information regarding the RoutePointDistributeCallRequest. The following type definition table lists the content of a RoutePointDistribute.

Parameter	Parameter Type	Required	Comments
<i>Agented</i>	SubscriberId	Yes	The agent subscriber ID.
<i>callingLineIdNumber</i>	Address	No	The calling line ID number to be displayed on the agent CPE.
<i>callingLineIdName</i>	String	No	The calling line ID name to be displayed on the agent CPE. This value overrides the value that would normally be supplied by Cisco BroadWorks.
<i>timeInQueue</i>	PositiveInteger	No	The amount of time (in seconds) that the caller has spent in queue waiting for an agent. This value is passed to the agent CPE through SIP protocol extensions. For more information, see section 8 Type Definitions .
<i>numberOfCallsInQueue</i>	PositiveInteger	No	The number of calls remaining in the Route Point queue. This value is passed to the agent CPE.
<i>longestWaitTime</i>	PositiveInteger	No	The longest wait time (in seconds) of the calls remaining in the Route Point queue. This value is passed to the agent CPE.
<i>whisperMessageAudioUrl</i>	String	No	The URL of the audio whisper message.
<i>whisperMessageVideoUrl</i>	String	No	The URL of the video whisper message.

8.8.14 RoutePointPlayTreatment

A RoutePointPlayTreatment provides information regarding the RoutePointPlayTreatmentRequest. The following type definition table lists the content of a RoutePointPlayTreatment.

Parameter	Parameter Type	Required	Comments
<i>audioUrlList</i>	RoutePointPlayTreatmentURLList	Yes	The list of audio URL to play.
<i>videoUrlList</i>	RoutePointPlayTreatmentURLList	No	The list of video URL to play.
<i>numberOfPlay</i>	RoutePointPlayTreatmentNumberOfPlay	No	Specifies the number of times to play the treatment. Can be a value between one and five. If not specified, the treatment is only played once.
<i>digitMap</i>	MGCPDigitMap	No	Digit map to define which digits to collect as specified in MGCP.

8.8.15 RoutePointPlayMusicOnHold

A RoutePointPlayMusicOnHold provides information regarding the RoutePointPlayTMusicOnHoldRequest. The following type definition table lists the content of a RoutePointPlayMusicOnHold.

Parameter	Parameter Type	Required	Comments
<i>duration</i>	PositiveInteger	No	The duration of the Music on Hold (MOH) to play (in seconds). If not present, the MOH is played without interruption.
<i>digitMap</i>	MGCPDigitMap	No	Digit map to define which digits to collect as specified in MGCP.
<i>playCollectId</i>	PlayCollectId	No	The playCollect Id used to identify a play-collect operation in progress. Only specified when extending the duration of the play-collect operation in progress. The <i>audioUrl</i> , <i>videoUrl</i> , and <i>mgcpDigitMap</i> parameters are ignored, only the new duration value is taken into consideration.

8.8.16 RoutePointPlayRingback

A RoutePointPlayRingback provides information regarding the RoutePointPlayTRingbackRequest. The following type definition table lists the content of a RoutePointPlayRingback.

Parameter	Parameter Type	Required	Comments
<i>duration</i>	PositiveInteger	No	The duration of the ringback in seconds.
<i>digitMap</i>	MGCPDigitMap	No	Digit map to define which digits to collect as specified in MGCP.
<i>playCollectId</i>	PlayCollectId	No	The playCollect Id used to identify a play-collect operation in progress. Only specified when extending the duration of the play-collect operation in progress. The <i>mgcpDigitMap</i> parameter is ignored, only the new duration value is taken into consideration.

8.8.17 RoutePointPlaySilence

A RoutePointPlaySilence provides information regarding the RoutePointPlaySilenceRequest. The following type definition table lists the content of a RoutePointPlaySilence.

Parameter	Parameter Type	Required	Comments
<i>duration</i>	PositiveInteger	No	The duration of the silence in seconds.
<i>digitMap</i>	MGCPDigitMap	No	Digit map to define which digits to collect as specified in MGCP.
<i>playCollectId</i>	PlayCollectId	No	The playCollect Id used to identify a play-collect operation in progress. Only specified when extending the duration of the play-collect operation in progress. The <i>mgcpDigitMap</i> parameter is ignored, only the new duration value is taken into consideration.

8.8.18 RoutePointPlayBusy

A RoutePointPlayBusy provides information regarding the RoutePointPlayBusyRequest. The following type definition table lists the content of a RoutePointPlayBusy.

Parameter	Parameter Type	Required	Comments
<i>duration</i>	PositiveInteger	No	The duration of the busy in seconds.
<i>busyPattern</i>	BusyPattern	Yes	Specify if a busy or fast busy must be played.
<i>playCollectId</i>	PlayCollectId	No	The playCollect Id used to identify a play-collect operation in progress. Only specified when extending the duration of the play-collect operation in progress. The mgcpDigitMap parameter is ignored, only the new duration value is taken into consideration.

8.8.19 RoutePointOutgoingDial

A RoutePointOutgoingDial provides information regarding the RoutePointOutgoingDialRequest. The following type definition table lists the content of a RoutePointOutgoingDial.

Parameter	Parameter Type	Required	Comments
<i>address</i>	Address	No	The address to use for the outgoing dial.
<i>routePointName</i>	String	No	The Route Point name. Cisco BroadWorks uses this name to determine the Calling Line Identification (CLID) settings for the outgoing call.

8.9 ACD

8.9.1 ACDQueue

An ACDQueue provides information about an ACD queue. The following type definition table lists the parameters of an ACDQueue.

Parameter	Parameter Type	Required	Comments
<i>queueEntries</i>	ACDQueueEntryList	No	The list of queue entries maintained in the ACD queue.

8.9.2 ACDQueueEntryList

A RoutePointQueueEntryList contains a list of ACDQueueEntry.

8.9.3 ACDQueueEntry

An ACDQueueEntry provides information about each call maintained in an ACD Queue. The following type definition table lists the parameters of an ACDQueueEntry.

Parameter	Parameter Type	Required	Comments
<i>callId</i>	CallId	Yes	The call ID of the call for the ACDQueueEntry.
<i>extTrackingId</i>	ExternalTrackingId	Yes	The external tracking ID of the call for the ACDQueueEntry.



Parameter	Parameter Type	Required	Comments
<i>remoteParty</i>	PartyInformation	Yes	Information about the remote party for the ACDQueueEntry.
<i>addTime</i>	Timestamp	Yes	The time when this entry was added to the ACD.
<i>removeTime</i>	Timestamp	No	The time when this entry was removed from the ACD. Only present when the entry has been removed from the queue.
<i>Mandatory Entrance</i>	EmptyContent	No	When present, indicates that the mandatory entrance is currently playing.
<i>bounced</i>	EmptyContent	No	When present, indicates that the call was offered to an agent and bounced.
<i>reordered</i>	EmptyContent	No	When present, indicates that the call was reordered within the queue.
<i>preservedWaitTime</i>	PositiveInteger	No	Indicates the preserved wait time in milliseconds of the call at the moment the call was actually added to the queue. The client application uses the attribute in conjunction to the addTime attribute to determine the actual wait time of the queued call. Only present if there is preserved wait time.
<i>ACDNumber</i>	Address	Yes	Indicates the ACD number the call was received on.
<i>ACDName</i>	String	Yes	Indicates the ACD name associated with the ACD number the call was received on.
<i>ACDPriority</i>	ACDPriorityType	Yes	The priority assigned to a call when inserted in the priority bucket in the queue. Present only if the ACD is of type Premium.
<i>addTimeInPriorityBucket</i>	Timestamp	No	The time this entry was added to the priority bucket in the queue. Present only if the Call Center is of type Premium.
<i>preservedWaitTimeInPriorityBucket</i>	PositiveInteger	No	Indicates the preserved wait time this entry has accumulated when being inserted in the priority bucket. The client application uses this value in conjunction with the addTimeInPriorityBucket to accurately display timestamps for the call in the priority bucket. Present only if the Call Center is of type Premium and there is a preserved wait time.
<i>AnsweringUserId</i>	UserId	No	The user ID of the agent that answered the call. This parameter is only present when the agent that answered the call is a Cisco BroadWorks subscriber.
<i>answeringCallId</i>	CallId	No	The call ID of the agent call when the call is answered by an agent. This parameter is only present when the call offer from the call center to the agent does not contain a SIP call leg. This parameter can be used to correlate with events generated for a call.

Parameter	Parameter Type	Required	Comments
<i>AnsweringNetworkCallId</i>	String	No	The SIP call ID of the agent call when the call is answered by an agent. This parameter is only present when the call offer from the call center to the agent contains a SIP call leg. This parameter can be used to correlate with events generated for a call.

8.9.3.1 ACDPriorityType

An ACDPriorityType specifies the various values for the priority assigned to a call inserted into an ACD queue. The following table provides the list of values that can be assigned to an ACDPriorityType.

ACDPriority	Description
0-Highest	Highest priority
1-High	High priority
2-Medium	Medium priority
3-Low	Low priority

8.9.4 ACDQueueEntryList

An ACDQueueEntryList contains a list of ACDQueueEntry.

8.9.5 QueueCallOrder

A QueueCallOrder maintains information about the position of queue entry in the ACD queue. The following type definition table lists the parameters of a QueueCallOrder.

Parameter	Parameter Type	Required	Comments
<i>Position</i>	PositiveInteger	Yes	The position of the queue entry.

8.9.6 ACDPromote

An ACDPromote provides information regarding an ACDPromoteCallRequest. The following type definition table lists the content of an ACDPromote.

Parameter	Parameter Type	Required	Comments
<i>Priority</i>	ACDPriorityType	Yes	The priority for the call's entry in the ACD queue.

8.9.7 CallCenterMonitoringStatus

A *CallCenterMonitoringStatus* provides information regarding the overall performance of an ACD.

Parameter	Parameter Type	Required	Comments
<i>averageHandlingTime</i>	NonNegativeIntWithSeverity	Yes	The average handling time for calls to this ACD.
<i>expectedWaitTime</i>	NonNegativeIntWithSeverity	No	The expected wait time for calls to this ACD. The expected wait time is a function of the average handle time and the number of agents in <i>Available</i> or <i>Wrap-up</i> state. Not present if there are no agents in the <i>Available</i> or <i>Wrap-up</i> state.
<i>averageSpeedOfAnswer</i>	NonNegativeIntWithSeverity	Yes	The average speed of answer for calls to this ACD.
<i>longestWaitTime</i>	NonNegativeIntWithSeverity	Yes	The longest wait time of any calls in the queue. The value reflects the longest wait time at the time the event was sent.
<i>numCallsInQueue</i>	NonNegativeIntWithSeverity	Yes	The number of calls in the queue.
<i>numAgentsAssigned</i>	NonNegativeInt	Yes	The number of agents assigned to the ACD.
<i>numAgentsStaffed</i>	NonNegativeInt	Yes	The number of agents staffing the ACD.
<i>numStaffedAgentsIdle</i>	NonNegativeInt	Yes	The number of staffed agents that are idle, that is, agents that are not on a call and that are in the <i>Available</i> state.
<i>numStaffedAgentsUnavailable</i>	NonNegativeInt	Yes	The number of staffed agents that are in the <i>Unavailable</i> state.

8.9.7.1 NonNegativeIntWithSeverity

A *NonNegativeIntWithSeverity* provides information on a measurement that can be assigned a severity-based value.

Parameter	Parameter Type	Required	Comments
<i>value</i>	NonNegativeInt	Yes	The measured value.

Parameter	Parameter Type	Required	Comments
<i>severity</i>	SimpleThresholdBasedSeverity	No	Severity is omitted if threshold values are not configured. Otherwise, it is always present.
<i>prevSeverity</i>	SimpleThresholdBasedSeverity	No	Previous alert severity is omitted if threshold values are not configured. It is omitted in initial responses to subscriptions. Otherwise, it is present whenever there has been a change in alert severity.
<i>threshold</i>	NonNegativeInt	No	This is the configured threshold value that determines the current severity. Threshold is omitted if threshold values are not configured. Otherwise, it is always present unless the current alert severity is "normal".

8.9.7.2 SimpleThresholdBasedSeverity

A SimpleThresholdBasedSeverity specifies the severity based on assigned threshold values. The following table provides the list of values that can be assigned to a SimpleThresholdBasedSeverity.

Agent State	Description
0	Normal
1	Higher than 0
2	Higher than 1

8.10 Agent

8.10.1 AgentStateInformation

An AgentStateInformation maintains information about the agent state. The following type definition table lists the parameters of an AgentStateInformation.

Parameter	Parameter Type	Required	Comments
<i>state</i>	AgentState	Yes	The current ACD agent state.
<i>stateTimestamp</i>	TimestampWithSeverity	Yes	Time stamp value of the last agent state change.
<i>unavailableCode</i>	AgentUnavailableCode	No	This element is present when the agent state is in the <i>Unavailable</i> state and when the unavailable codes are enabled.

Parameter	Parameter Type	Required	Comments
<i>signInTimestamp</i>	Timestamp	No	Present when the agent is not in the <i>Sign-out</i> state. This is the time stamp value of the agent sign-in (that is, transition out the <i>Sign-out</i> state).
<i>totalAvailableTime</i>	NonNegativeInt	No	Present when the agent is not in the <i>Sign-out</i> state. This is the total available time since the last agent sign-in. Time currently spent in the <i>Available</i> state not included.
<i>averageWrapUpTime</i>	NonNegativeIntWithSeverity	No	Present when the agent is not in the <i>Sign-out</i> state. This is the average wrap-up time per handled call since the last sign-in.
<i>wrapUpCallId</i>	CallId	No	This element is present when the agent state is in the <i>Wrap-up</i> state. It provides the call ID of the last ACD or Route point call the agent was involved in.
<i>wrapUpCallCenterUserId</i>	UserId	No	This element is present when the agent is in the <i>Wrap-up</i> state. It indicates the user ID of the ACD for the last ACD or outgoing ACD call made by the agent. This information element can be used to determine the appropriate list of disposition codes that can be entered by the agent in <i>Wrap-up</i> .

8.10.1.1 AgentState

An AgentState specifies the state of an agent. The following table provides the list of values that can be assigned to an AgentState.

Agent State	Description
Sign-In	Agent is signing in. This state is a temporary state while logging in.
Available	Agent is available.
Unavailable	Agent is unavailable.
Wrap-Up	Agent is performing post-call work.
Sign-out	Agent is signed out.

8.10.1.2 TimestampWithSeverity

A *TimestampWithSeverity* provides information on a timestamp that can be assigned a severity-based value.

Parameter	Parameter Type	Required	Comments
<i>value</i>	Timestamp	Yes	The timestamp value.
<i>severity</i>	SimpleThresholdBased Severity	No	Severity is omitted if threshold values are not configured. Otherwise, it is always present.
<i>prevSeverity</i>	SimpleThresholdBased Severity	No	Previous alert severity is omitted if threshold values are not configured. It is omitted in initial responses to subscriptions. Otherwise, it is present whenever there is a change in alert severity.
<i>threshold</i>	NonNegativeInt	No	This is the configured threshold value that determines the current severity. Note that for a timestamp, this threshold is compared to the difference between the timestamp "value" and the current time. It is in units of seconds. Threshold is omitted if threshold values are not configured. Otherwise, it is always present unless the current alert severity is "normal".

8.10.2 AgentUnavailableCode

An AgentUnavailableCode maintains an agent unavailable code. An AgentUnavailableCode is represented with a string.

8.10.3 AgentDispositionCode

An AgentDispositionCode maintains a call disposition code. The AgentDispositionCode is represented with a string.

8.10.4 ACDAgentJoinUpdateData

An ACDAgentJoinUpdateData contains the ACD service update data for agent joining or leaving. The following type definition table lists the parameters of an ACDAgentJoinUpdateData.

Parameter	Parameter Type	Required	Comments
<i>joinInfo</i>	ACDAgentJoinInfo	Yes	The ACD Agent Join Info indicating that the agent is joining or leaving an ACD. Only not present for service unassignment notifications.

8.10.5 ACDAgentJoinInfo

An ACDAgentJoinInfo contains information about an ACD an agent belongs to and may be joined to. The following type definition table lists the parameters of an ACDAgentJoinInfo.

Parameter	Parameter Type	Required	Comments
<i>acdUserId</i>	userId	Yes	The ACD user ID.
<i>joined</i>	EmptyContent	No	Only present when the agent has joined the ACD.
<i>skillLevel</i>	SkillLevel	No	Only present when the agent is in Skill-Based ACD.

8.10.5.1 SkillLevel

A SkillLevel maintains information about the skill level of the agent in the queue. It is used if the queue uses Skill-Based Routing. The SkillLevel is represented with a PositiveInteger.

8.10.6 ACDAgentJoinData

An ACDAgentJoinData contains the Call Center service data that an agent has joined to or belongs to. The following type definition table lists the parameters of an ACDAgentJoinData.

Parameter	Parameter Type	Required	Comments
<i>joinInfos</i>	ACDAgentJoinInfoList	No	List of ACD Agent Join Info. Only present when the agent belongs to at least one ACD.

8.10.7 ACDAgentJoinInfoList

An ACDAgentJoinInfoList contains a list of ACDAgentJoinInfo.

8.10.8 Call Center

A CallCenter contains an agent call center settings. The following type definition table lists the parameters of a CallCenter.

Parameter	Parameter Type	Required	Comments
<i>agentACDState</i>	AgentACDState	Yes	The state of the agent.
<i>unavailableCode</i>	AgentUnavailableCode	No	This element is present when the agent state changes to the <i>Unavailable</i> state and when the unavailable codes are enabled.
<i>useDefaultGuardTimer</i>	Boolean	No	Not used by CTI interface.
<i>enableGuardTimer</i>	Boolean	No	Not used by CTI interface.
<i>guardTimerSeconds</i>	ACDGuardTimerSeconds		Not used by CTI interface.
<i>useSystemDefaultUnavailableSettings</i>	Boolean	No	Not used by CTI interface.
<i>forceAgentUnavailableOnDNDActivation</i>	Boolean	No	Not used by CTI interface.

Parameter	Parameter Type	Required	Comments
<i>forceAgentUnavailableOnBouncedCallLimit</i>	Boolean	No	Not used by CTI interface.
<i>numberConsecutiveBouncedCallsToForceAgentUnavailable</i>	ACDConsecutiveBouncedCallsToForceAgentUnavailable	No	Not used by CTI interface.
<i>ACDConsecutiveBounceCallsToForceAgentUnavailable</i>	Boolean	No	Not used by CTI interface.
<i>makeOutgoingCallsAsCallCenter</i>	DNISKey	No	Not used by CTI interface.

8.10.9 CallCenterAgentMonitoringStatus

A *CallCenterAgentMonitoringStatus* provides information regarding the overall performance of an agent.

Parameter	Parameter Type	Required	Comments
<i>averageACDCALLTime</i>	NonNegativeIntWithSeverity	No	The average call time for ACD calls since the last sign-in. Not present when the agent state is <i>Sign-out</i> .
<i>averageACDOutgoingCallTime</i>	NonNegativeIntWithSeverity	No	The average call time for outgoing ACD calls since the last sign-in. Not present when the agent state is <i>Sign-out</i>

8.10.10 CallCenterAgentOnCallAlertInfo

The *CallCenterAgentOnCallAlertInfo* indicates if an agent is:

- on a call or idle
- for how long, for both cases
- whether that length of time is greater than a configured threshold

Parameter	Parameter Type	Required	Comments
<i>alertEvent</i>	String	Yes	Valid values are "OnCallTime" and "IdleTime".
<i>timestamp</i>	TimestampWithSeverity	Yes	Indicates for how long the agent was on a call or idle.

8.11 Meet-Me Conference Related Types

8.11.1 MeetMeConference

A *MeetMeConference* provides information regarding the state of the conference including the list of participants and moderators.

Parameter	Parameter Type	Required	Comments
<i>conferenceInfo</i>	MeetMeConferenceInfo	Yes	Information about the conference state.
<i>conferenceParticipants</i>	MeetMeConferenceParticipants	Yes	The details of the participants in the conference.

8.11.1.1 MeetMeConferenceInfo

A *MeetMeConferenceInfo* provides the details of an instance of an active Meet-Me conference.

Parameter	Parameter Type	Required	Comments
<i>lectureMode</i>	EmptyContent	No	Present when the lecture mode is enabled.
<i>automaticLectureMode</i>	EmptyContent	No	Present when the automatic lecture mode is enabled.
<i>lock</i>	EmptyContent	No	Present when the conference is locked.
<i>recordingState</i>	MeetMeConferenceRecordingState	Yes	Defines the recording state of the conference
<i>securityClassification</i>	NonEmptyToken	No	The conference security classification level. Present if the conference owner has the Security Classification service assigned.
<i>clientSessionInfo</i>	ClientSessionInfo	No	An opaque token set by the client. Present if client session info has been set for this conference.

8.11.1.1.1 MeetMeConferenceRecordingState

A *MeetMeConferenceRecordingState* specify the recording state of a Meet-Me conference. The following table provides the list of values that can be assigned to a *MeetMeConferenceRecordingState*.

State	Description
Not Started	The recording has not been started.
Recording	The conference is being recorded.
Pause	The recording has been paused.
End	The recording has ended.

8.11.1.2 MeetMeConferenceParticipants

A *MeetMeConferenceParticipants* contains a list of Meet-Me conference participants.

8.11.1.3 MeetMeConferenceParticipant

A *MeetMeConferenceParticipant* provides the details of a Meet-Me conference participant. The following table lists the parameters of a *MeetMeConferenceParticipant*.

Parameter	Parameter Type	Required	Comments
<i>callId</i>	CallId	Yes	The call ID of the call for the participant.
<i>partyInformation</i>	PartyInformation	Yes	Information about the participant on the conference.
<i>activeTalker</i>	EmptyContent	No	When present, indicates that the active participant is the active talker on the conference.
<i>held</i>	EmptyContent	No	When present, indicates that the participant has been put on hold by a host.
<i>moderator</i>	EmptyContent	No	When present, indicates that the participant is a moderator in the conference.
<i>muted</i>	EmptyContent	No	When present, indicates that the participant has been muted by a host.
<i>uniqueIdentifier</i>	MeetMeConferenceUniqueIdentifier	No	Occurs when a participant has keyed a self-identifying pin when joining the conference.

8.11.1.3.1 MeetMeConferenceUniqueIdentifier

A *MeetMeConferenceUniqueIdentifier* contains the self-identifying pin entered by the participant when joining the conference. A *MeetMeConferenceUniqueIdentifier* is represented with a string.

8.12 Collaborate Room Related Types

8.12.1 CollaborateRoom

A *CollaborateRoom* provides information regarding the state of the room including the list of participants.

Parameter	Parameter Type	Required	Comments
<i>roomSessionInfo</i>	CollaborateRoomSessionInfo	Yes	Information about the room state.
<i>roomParticipants</i>	CollaborateRoomParticipants	Yes	The details of the participants in the room.

8.12.1.1 CollaborateRoomSessionInfo

A *CollaborateRoomSessionInfo* provides the details of an instance of an active Collaborate Room.

Parameter	Parameter Type	Required	Comments
<i>roomType</i>	CollaborateRoomType	Yes	Indicates the type of room.
<i>roomId</i>	NonEmptyToken	Yes	Indicates the room Id.
<i>locked</i>	EmptyContent	No	Present when the room is locked.
<i>clientSessionInfo</i>	ClientSessionInfo	No	An opaque token set by the client. Present if client session info has been set for this conference.

8.12.1.2 CollaborateRoomParticipants

A *CollaborateRoomParticipants* contains a list of Collaborate Room participants.

8.12.1.3 CollaborateRoomParticipant

A *CollaborateRoomParticipant* provides the details of a Collaborate Room participant. The following table lists the parameters of a *CollaborateRoomParticipant*.

Parameter	Parameter Type	Required	Comments
<i>callId</i>	CallId	Yes	The call ID of the call for the participant.
<i>partyInformation</i>	PartyInformation	Yes	Information about the participant in the room.
<i>activeTalker</i>	EmptyContent	No	When present, indicates that the active participant is the active talker in the room.
<i>owner</i>	EmptyContent	No	When present, indicates that the participant is the owner of the room.

Parameter	Parameter Type	Required	Comments
<i>muted</i>	EmptyContent	No	When present, indicates that the participant has been muted.

8.12.1.4 CollaborateRoomType

A *CollaborateRoomType* specifies the type of room. The following table provides the list of values that can be assigned to a *CollaborateRoomType*.

EntityType	Description
MyRoom	The room that is assigned by default to the subscriber.
InstantRoom	The room that is temporarily created by the subscriber for immediate use.
ProjectRoom	The room that is created by the subscriber for special purpose projects. These rooms may be created for one-time or schedulable sessions.

8.13 Service Management

8.13.1 CallForwardingAlways

A *CallForwardingAlways* maintains information on the configuration for the Call Forwarding Always service. The following type definition table lists the content of a *CallForwardingAlways*.

Parameter	Parameter Type	Required	Comments
<i>isActive</i>	Boolean	No	When set to true, indicates that CFA is enabled. When set to false, indicates that CFA is disabled.
<i>forwardToPhoneNumber</i>	Address	No	Address of the forwarded destination. Only used if CFA is enabled.
<i>ringSplash</i>	Boolean	No	When set to true, indicates that the subscriber should be notified through a ring tone that a call has been forwarded. Only used if CFA is enabled.

8.13.1.1 OutgoingDNorSIPURI

An *OutgoingDNorSIPURI* is a Phone Number or SIP URI that can be used to dial. The *OutgoingDNorSIPURI* is represented with a string.

8.13.2 CallForwardingNoAnswer

A *CallForwardingNoAnswer* maintains information on the configuration for the Call Forwarding No Answer service. The following type definition table lists the content of a *CallForwardingNoAnswer*.

Parameter	Parameter Type	Required	Comments
<i>isActive</i>	Boolean	No	When set to true, indicates that CFNA is enabled. When set to false, indicates that CFNA is disabled.
<i>forwardToPhoneNumber</i>	Address	No	Address of the forwarded destination. Only used if CFNA is enabled.

Parameter	Parameter Type	Required	Comments
<i>numberOfRings</i>	NumberOfRingsZeroToTwentyExcludeOne	No	The number of ring to activate CFNA.

8.13.2.1 NumberOfRingsZeroToTwentyExcludeOne

A NumberOfRingsZeroToTwentyExcludeOne specifies the number of rings to active CFNA. The NumberOfRingsZeroToTwentyExcludeOne is represented with an int that must be a number between 0 and 20 inclusively (with the exclusion of 1).

8.13.3 CallForwardingBusy

A CallForwardingBusy type maintains information on the configuration for the Call Forwarding Busy service. The following type definition table lists the content of a CallForwardingBusy.

Parameter	Parameter Type	Required	Comments
<i>isActive</i>	Boolean	No	When set to true, indicates that CFB is enabled. When set to false, indicates that CFB is disabled.
<i>forwardToPhoneNumber</i>	Address	No	Address of the forwarded destination. Only used if CFB is enabled.

8.13.4 DoNotDisturb

A DoNotDisturb maintains information about a DND configuration. The following type definition table lists the content of a DoNotDisturb.

Parameter	Parameter Type	Required	Comments
<i>isActive</i>	Boolean	No	When set to true, indicates that DND is enabled. When set to false, indicates that DND is disabled.
<i>ringSplash</i>	Boolean	No	When set to true, indicates that the subscriber should be notified through a ring tone that the DND service is exercised. Only used if DND is enabled.

8.13.5 BroadWorksAnywhereInfo

A BroadWorksAnywhereInfo maintains information about a BroadWorksAnywhereInfo configuration. The following type definition table lists the content of a BroadWorksAnywhereInfo.

Parameter	Parameter Type	Required	Comments
<i>alertAllLocationsForClickToDialCalls</i>	Boolean	No	When true, indicates that all active locations should be alerted for Click-to-Dial calls.
<i>locations</i>	BroadWorksAnywhereLocationInfoList	No	This defines the locations associated with BroadWorks Anywhere. This is present when at least one location is set in the BroadWorks Anywhere service.

8.13.5.1 BroadWorksAnywhereLocationInfoList

A BroadWorksAnywhereLocationInfoList contains a list of BroadWorksAnywhereLocationInfo.

8.13.5.2 BroadWorksAnywhereLocationInfo

A BroadWorksAnywhereLocationInfo maintains information about a BroadWorks Anywhere location. The following type definition table lists the content of a BroadWorksAnywhereLocationInfo.

Parameter	Parameter Type	Required	Comments
<i>locationUri</i>	URL	No	Not used by the CTI interface.
<i>phoneNumber</i>	Address	No	The number of old messages. It is only present when there are old messages.
<i>active</i>	Boolean	No	When present, indicates this BroadWorks Anywhere location is enabled.
<i>description</i>	NonEmptyToken	No	The description for a BroadWorks Anywhere location.
<i>criteriaActivations</i>	CriteriaActivationList	No	The criteria set for the location. This selective criteria define which calls can be routed to this location and when. This is present when at least one criterion is set.

8.13.5.3 CriteriaActivationList

A CriteriaActivationList contains a list of CriteriaActivation.

8.13.5.4 CriteriaActivation

A CriteriaActivation maintains information about a BroadWorks Anywhere activation criterion. The following type definition table lists the content of a CriteriaActivation.

Parameter	Parameter Type	Required	Comments
<i>criteriaName</i>	String	Yes	Indicates the name of the criteria.
<i>active</i>	Boolean	No	When present, indicates the criteria active.
<i>Uri</i>	URL	No	The URI for accessing the other details of this criterion.

8.13.6 ActivatableServiceInfo

An ActivatableServiceInfo indicates whether a service is active. The following type definition table lists the content of an ActivatableServiceInfo.

Parameter	Parameter Type	Required	Comments
<i>active</i>	Boolean	Yes	Indicates the service is enabled or not.



8.13.7 CallTransferInfo

A CallTransferInfo maintains information about the Call Transfer service configuration data. The following type definition table lists the content of a CallTransferInfo.

Parameter	Parameter Type	Required	Comments
<i>recallActive</i>	Boolean	No	Only present when Call Transfer Recall is enabled.
<i>recallNumberOfRings</i>	NumberOfRingsTwoToTwenty	No	Number of rings before Call Transfer Recall is invoked. Only present when Call Transfer Recall is enabled.
<i>busyCampOnActive</i>	Boolean	No	Only present when Busy Camp On is enabled.
<i>busyCampOnSeconds</i>	CallTransferBusyCampOnSeconds	No	Number of seconds for Busy Camp On timer. Only present when Busy Camp On is enabled.
<i>useDiversionInhibitorForBlindTransfer</i>	Boolean	No	Only present when the diversion inhibitor is enabled for blind transfers.
<i>useDiversionInhibitorForConsultativeCalls</i>	Boolean	No	Only present when the diversion inhibitor is enabled for consultative calls.

8.13.7.1 NumberOfRingsTwoToTwenty

The NumberOfRingsTwoToTwenty is an Integer from 2 through 20.

8.13.7.2 CallTransferBusyCampOnSeconds

The CallTransferBusyCampOnSeconds is an Integer from 30 through 600.

8.13.8 FlexibleSeatingGuest

A FlexibleSeatingGuest maintains information about a Flexible Seating Guest configuration. The following type definition table lists the content of a FlexibleSeatingGuest.

Parameter	Parameter Type	Required	Comments
<i>active</i>	Boolean	Yes	Indicates if the service is enabled or not.
<i>enableAssociationLimit</i>	Boolean	Yes	Indicates if the association limit is enabled or not.
<i>associationLimitHours</i>	FlexibleSeatingAssociationLimitHours	Yes	Indicates the duration of the association limit.
<i>unlockPhonePINCode</i>	FlexibleSeatingGuestUnlockPhonePINCode	No	Present when the PIN code for unlocking the phone has been provisioned.
<i>accessDevice</i>	FlexibleSeatingGuestAccessDevice	No	Present when the guest access device has been provisioned.
<i>hostUserId</i>	UserId	No	Present when the guest has been associated with the host.
<i>hostLastName</i>	LastName	No	Not used by the CTI interface.

Parameter	Parameter Type	Required	Comments
<i>hostFirstName</i>	FirstName	No	Not used by the CTI interface.
<i>hostEnforcesAssociationLimit</i>	Boolean	No	Not used by the CTI interface.
<i>hostAssociationLimitHours</i>	FlexibleSeatingAssociationLimitHours	No	Not used by the CTI interface.

8.13.8.1 FlexibleSeatingAssociationLimitHours

A FlexibleSeatingAssociationLimitHours indicates Maximum time limit for a flexible seating guest's association to flexible seating hosts. It is maintained using an Integer. The range of values that can be assigned to an Integer is from 1 through 999.

8.13.8.2 FlexibleSeatingGuestUnlockPhonePINCode

A FlexibleSeatingGuestUnlockPhonePINCode maintains the unlock phone PIN code for the flexible seating guest. It is stored as a String.

8.13.8.3 FlexibleSeatingGuestAccessDevice

A FlexibleSeatingGuestAccessDevice maintains information about the device associated with a flexible seating guest. The following type definition table lists the content of a FlexibleSeatingGuestAccessDevice.

Parameter	Parameter Type	Required	Comments
<i>deviceName</i>	AccessDeviceName	Yes	Access device name.
<i>deviceLevel</i>	AccessDeviceLevel	Yes	Identifies at which level in the system an identity/device profile is created.
<i>deviceLinePort</i>	AccessDeviceEndpointLinePort	No	The Line/Port identifies a device endpoint in standalone mode or a SIPURI public identity in IMS mode. Only present if provisioned.

8.13.8.3.1 AccessDeviceName

An AccessDeviceName maintains the name of the access device. An AccessDeviceName is represented with a string.

8.13.8.3.2 AccessDeviceLevel

An AccessDeviceLevel identifies at which level in the system an identity/device profile is created. The following table provides the list of values that can be assigned to an AccessDeviceLevel.

State	Description
System	System level access device.
Service Provider	Service Provider level access device.
Group	Group level access device.

8.13.8.3.3 AccessDeviceEndpointLinePort

An AccessDeviceEndpointLinePort maintains the address of record that identifies a device endpoint in standalone mode or a SIPURI public identity in IMS mode. An AccessDeviceEndpointLinePort is represented with a string.

8.13.9 SequentialRing

A SequentialRing maintains information about the Sequential Ring configuration data. The following type definition table lists the content of a SequentialRing.

Parameter	Parameter Type	Required	Comments
<i>active</i>	Boolean	No	Indicates whether the service is enabled or not.
<i>ringBaseLocationFirst</i>	Boolean	No	Indicates whether the base location should be rung first.
<i>baseLocationNumberOfRings</i>	NumberOfRings ZeroToTwenty ExcludeOne	No	The number of rings to ring the base location.
<i>continueIfBaseLocationIsBusy</i>	Boolean	No	Indicates the need to continue the search process when the base location is busy.
<i>callerMayStopSearch</i>	Boolean	No	Indicates that the caller can skip the search process. This assumes that forwarding or messaging is enabled.
<i>location1</i>	SequentialRing Location	No	The first sequential ring location.
<i>location2</i>	SequentialRing Location	No	The second sequential ring location.
<i>location3</i>	SequentialRing Location	No	The third sequential ring location.
<i>location4</i>	SequentialRing Location	No	The fourth sequential ring location.
<i>location5</i>	SequentialRing Location	No	The fifth sequential ring location.
<i>CriteriaActivationList</i>	CriteriaActivation List	No	The list of criteria for Sequential Ring. This is present what at least one criterion is set.

8.13.9.1 SequentialRingLocation

The SequentialRingLocation contains the information for a Sequential Ring location. The following type definition table lists the content of a SequentialRingLocation.

Parameter	Parameter Type	Required	Comments
<i>Address</i>	Address	No	The address for this Sequential Ring location. Only present when an address has been set.
<i>numberOfRings</i>	NumberOfRings ZeroToTwenty ExcludeOne	No	The number of rings to ring the specified address.

Parameter	Parameter Type	Required	Comments
<i>Answer Confirmation Required</i>	Boolean	No	Indicates whether answer confirmation is required for the location.

8.13.10 SimultaneousRingPersonal

A SimultaneousRingPersonal maintains information about the Simultaneous Ring Personal configuration data. The following type definition table lists the content of a SimultaneousRingPersonal.

Parameter	Parameter Type	Required	Comments
<i>active</i>	Boolean	No	Indicates whether the service is enabled or not.
<i>incomingCalls</i>	SimultaneousRing Selection	No	Setting for how incoming calls are handled by the Simultaneous Ring Personal service.
<i>simRing Locations</i>	SimultaneousRing LocationList	No	The locations configured for the Simultaneous Ring Personal service. Only present when at least one location has been set.
<i>Criteria ActivationList</i>	CriteriaActivation List	No	The list of criteria for Simultaneous Ring Personal. This is present what at least one criterion is set.

8.13.10.1 SimultaneousRingSelection

The SimultaneousRingSelection specifies how the Simultaneous Ring Personal service handles incoming calls, specifically in cases where the user is already busy on a call.

Selection	Description
Do not Ring if on a Call	If the user is busy on a call, then the secondary locations are not alerted.
Ring for all Incoming Calls	If the user is busy on a call, then the secondary locations are still alerted.

8.13.10.2 SimultaneousRingLocationList

A SimultaneousRingLocationList contains a list of SimultaneousRingLocation.

8.13.10.3 SimultaneousRingLocation

The SimultaneousRingLocation contains the information for a Simultaneous Ring location. The following type definition table lists the content of a SimultaneousRingLocation.

Parameter	Parameter Type	Required	Comments
<i>Address</i>	Address	Yes	The address for the location.
<i>answerConfirmation Required</i>	Boolean	No	Indicates whether answer confirmation is required for the location.

8.13.11 VoiceMailMessageSummary

A VoiceMailMessageSummary maintains summary information about the voice mail content. The following type definition table lists the content of a VoiceMailMessageSummary.

Parameter	Parameter Type	Required	Comments
<i>summary</i>	MessageSummary	Yes	The updated voice message summary for the Third-Party Voice Mail Support/Voice Messaging service.

8.13.11.1 MessageSummary

A MessageSummary maintains summary information about the voice mail content. The following type definition table lists the content of a MessageSummary.

Parameter	Parameter Type	Required	Comments
<i>newMessages</i>	PositiveInt	No	The number of new messages. Only present when there are new messages.
<i>oldMessages</i>	PositiveInt	No	The number of old messages. Only present when there are old messages.
<i>newUrgentMessages</i>	PositiveInt	No	The number of new urgent messages. Only present when there are new urgent messages.
<i>oldUrgentMessages</i>	PositiveInt	No	The number of old urgent messages. Only present when there are old urgent messages.

8.13.12 Remote Office

A RemoteOffice maintains information about the Remote Office service configuration data. The following type definition table lists the content of a RemoteOffice.

Parameter	Parameter Type	Required	Comments
<i>Active</i>	Boolean	No	When present, indicates the service is enabled. When not present, service is inactive.
<i>remoteOfficeNumber</i>	Address	No	The Remote Office address. This is present when the service is enabled.

8.13.13 CommPilotExpress

A CommPilotExpress maintains information about the CommPilot Express service configuration data. The following type definition table lists the content of a CommPilotExpress.

Parameter	Parameter Type	Required	Comments
<i>profile</i>	CommPilotExpress Profile	No	Specifies the CommPilot Express profile setting.
<i>availableInOffice</i>	CommPilotExpress AvailableInOffice	No	The “In-Office” configuration.
<i>availableOutOfOffice</i>	CommPilotExpress AvailableOutOfOffice	No	The “Out-of-Office” configuration.



Parameter	Parameter Type	Required	Comments
<i>busy</i>	CommPilotExpressBusy	No	The “Busy” configuration.
<i>unavailable</i>	CommPilotExpressUnavailable	No	The “Unavailable” configuration.

8.13.13.1 CommPilotExpressProfile

A CommPilotExpressProfile specifies the names of the CommPilot Express profile. The following table provides the list of values that can be assigned to a CommPilotExpressProfile.

Agent State	Description
None	No profile
Available In Office	The available and in the office profile.
Available Out Of Office	The available but out of the office profile.
Busy	The busy profile.
Unavailable	The unavailable profile.

8.13.13.2 CommPilotExpressAvailableInOffice

A CommPilotExpressAvailableInOffice maintains information about the CommPilot Express Available In-Office configuration data. The following type definition table lists the content of a CommPilotExpressAvailableInOffice.

Parameter	Parameter Type	Required	Comments
<i>additionalPhoneNumberToRing</i>	Address	No	An additional number or SIP URI to ring. This is present when the additional address is set.
<i>busySetting</i>	CommPilotExpressRedirection	No	The redirection behavior when the user receives a call while in the office and busy.
<i>noAnswerSetting</i>	CommPilotExpressRedirection	No	The redirection behavior when the user receives a call while in the office and not answering.

8.13.13.3 CommPilotExpressRedirection

A CommPilotExpressRedirection maintains information about the redirection type specified by the CommPilot Express configuration. The following type definition table lists the content of a CommPilotExpressRedirection.

Parameter	Parameter Type	Required	Comments
<i>action</i>	CommPilotExpressRedirectionAction	No	The action to take when redirecting.
<i>forwardingAddress</i>	Address	No	The forwarding address. This is only valid when the action is set to “Forward”.

8.13.13.3.1 CommPilotExpressRedirectionAction

A CommPilotExpressRedirectionAction specifies the names of the CommPilot Express profile. The following table provides the list of values that can be assigned to a CommPilotExpressRedirectionAction.

Agent State	Description
Transfer To Voice Mail	Transfer the call to voice mail.
Forward	Forward the call to the specified destination.

8.13.13.4 CommPilotExpressAvailableOutOfOffice

A CommPilotExpressAvailableOutOfOffice maintains information about the CommPilot Express Available Out-Of-Office configuration data. The following type definition table lists the content of a CommPilotExpressAvailableOutOfOffice.

Parameter	Parameter Type	Required	Comments
<i>incomingCalls</i>	CommPilotExpress Redirection	No	The redirection behavior when out of the office but available to take calls.
<i>incomingCallNotify</i>	CommPilotExpress EmailNotify	No	The e-mail address and notification setting for notifying an incoming call.

8.13.13.5 CommPilotExpressEmailNotify

A CommPilotExpressEmailNotify maintains information about the CommPilot Express E-mail Notify configuration. The following type definition table lists the content of a CommPilotExpressEmailNotify.

Parameter	Parameter Type	Required	Comments
<i>sendEmail</i>	Boolean	No	When present and true, indicates an e-mail should be sent.
<i>emailAddress</i>	EmailAddress	No	The e-mail address to which to send the voice message notification. Present when the e-mail address has been set.

8.13.13.5.1 EmailAddress

An EmailAddress contains an email address. An EmailAddress is represented with a string.

8.13.13.6 CommPilotExpressBusy

A CommPilotExpressBusy maintains information about a CommPilot Express Busy configuration data. The following type definition table lists the content of a CommPilotExpressBusy.

Parameter	Parameter Type	Required	Comments
<i>incomingCalls</i>	CommPilotExpress RedirectionWith Exception	No	The settings for e-mail notification of voice messages.
<i>voiceMailNotify</i>	CommPilotExpress EmailNotify	No	The redirection behavior when the user receives a call while in the office and busy.

8.13.13.6.1 CommPilotExpressRedirectionWithException

A CommPilotExpressRedirectionWithException maintains information about the type of transfer to voice mail or forward to a number with certain exceptions. The following type definition table lists the content of a CommPilotExpressBusy.

Parameter	Parameter Type	Required	Comments
<i>sendCallsToVoiceMailExceptExcludedAddresses</i>	Boolean	No	When present, indicates the need to send calls to Voice Messaging on all calls except for excluded numbers.
<i>ExcludedAddress1</i>	Address	No	Only valid when <i>sendCallToVoiceMailExceptExcludeNumber</i> is present.
<i>ExcludedAddress2</i>	Address	No	Only valid when <i>sendCallToVoiceMailExceptExcludeNumber</i> is present.
<i>ExcludedAddress3</i>	Address	No	Only valid when <i>sendCallToVoiceMailExceptExcludeNumber</i> is present.

8.13.13.7 CommPilotExpressUnavailable

A CommPilotExpressUnavailable maintains information about a CommPilot Express Unavailable configuration data. The following type definition table lists the content of a CommPilotExpressUnavailable.

Parameter	Parameter Type	Required	Comments
<i>incomingCalls</i>	CommPilotExpressRedirectionWithException	No	The settings for e-mail notification of voice messages.
<i>voicemailGreeting</i>	CommPilotExpressVoiceMailGreeting	No	The type of Voice Mail greeting to be used.

8.13.13.7.1 CommPilotExpressVoiceMailGreeting

A CommPilotExpressVoiceMailGreeting specifies the Voice Messaging Greeting Choices of the CommPilot Express profile. The following table provides the list of values that can be assigned to a CommPilotExpressVoiceMailGreeting.

Agent State	Description
No Answer	No answer greeting.
Unavailable	Unavailable greeting.

8.13.14 HotelingGuest

A HotelingGuest maintains information about the guest configuration and association. The following type definition table lists the content of a HotelingGuest.

Parameter	Parameter Type	Required	Comments
<i>isActive</i>	Boolean	No	When set to true, indicates that the subscriber can be associated with a host subscriber account. If set to false and the guest subscriber is associated with a host account, then this subscriber is forcibly disassociated.
<i>enableAssociationLimit</i>	Boolean	No	When set to true, indicates that the association time limit is enforced.
<i>associationLimitHours</i>	HotelingAssociationLimitHours	No	The timeout value in hours the subscriber can configure when associating with the host. It can be set to an integer value greater than 0 and less than or equal to 999. The guest timeout setting is a timer that will automatically disassociate the guest from the host after the specified period of time.
<i>hostUserId</i>	UserId	No	The host currently associated with the user.

8.13.14.1 HotelingAssociationLimitHours

A HotelingAssociationLimitHours specifies the maximum time limit for a hoteling guest's association to hoteling hosts. The HotelingAssociationLimitHours is represented with a PositiveInteger and must be smaller than 999 inclusively.

8.13.15 AvailableHotelngHosts

An AvailableHotelngHosts maintains information about the list of available Hoteling Host for a Guest. The following type definition table lists the content of an AvailableHotelngHosts.

Parameter	Parameter Type	Required	Comments
<i>hostUserIdList</i>	UserIdList	Yes	List of user IDs.

8.13.16 UserIdList

The userIdList contains a list of user IDs.

8.13.17 PersonalAssistant

A PersonalAssistant maintains information about a Personal Assistant configuration. The following type definition table lists the content of a PersonalAssistant.

Parameter	Parameter Type	Required	Comments
<i>presence</i>	PersonalAssistantPresence	Yes	Indicates if the service is enabled or not.
<i>enableExpirationTime</i>	Boolean	Yes	Indicates if the current presence has an expiration time.
<i>expirationTime</i>	DateTime	Yes	Indicates the date and time at which the current presence expires. Upon expiry, the presence is reset to "None". The data and time is in the format "YYYY-MM-DDThh:mm:ss".

8.13.17.1 PersonalAssistantPresence

A *PersonalAssistantPresence* identifies the presence status currently active for the subscriber. The following table provides the list of values that can be assigned as *PersonalAssistantPresence*.

Presence	Description
<i>None</i>	Indicates that there is no presence status currently active for the subscriber.
<i>Business Trip</i>	Subscriber is on a business trip.
<i>Gone For The Day</i>	Subscriber is gone for the day.
<i>Lunch</i>	Subscriber is at lunch.
<i>Meeting</i>	Subscriber is in a meeting.
<i>Out Of Office</i>	Subscriber is out of the office.
<i>Temporarily Out</i>	Subscriber is temporarily out.
<i>Training</i>	Subscriber is in training.
<i>Unavailable</i>	Subscriber is unavailable.
<i>Vacation</i>	Subscriber is on vacation.

8.14 Application Controller

An ApplicationController maintains information about an application Controller. The following type definition table lists the content of an ApplicationController.

Parameter	Parameter Type	Required	Comments
<i>State</i>	ApplicationControllerState	Yes	The state of the application Controller.

8.14.1 ApplicationControllerState

An ApplicationControllerState specifies the state of an application controller. The following table provides the list of values that can be assigned to an ApplicationControllerState.

Agent State	Description
ready	The application controller <i>ready</i> state.
not ready	The application controller <i>not ready</i> state.

8.15 Voice Messaging

8.15.1 VoiceMessagingMessages

A VoiceMessagingMessages contains a list of any messages currently present in the user's voice mailbox. The following type definition table lists the content of a VoiceMessagingMessages.

Parameter	Parameter Type	Required	Comments
<i>messageInfoList</i>	MessageInfoList	Yes	A list of information about each of the user's voice messages. If no messages are present for the user, the list is empty.

8.15.1.1 MessageInfoList

A MessageInfoList contains a list of MessageInfo.

8.15.1.2 MessageInfo

A MessageInfo contains detailed summary information for an individual voice message. The following type definition table lists the content of a MessageInfo.

Parameter	Parameter Type	Required	Comments
<i>duration</i>	NonNegativeInt	No	The duration of the message, in milliseconds. Not present for fax messages.
<i>callingPartyInfo</i>	VoiceMailPartyInformation	No	The information of the calling party who left the message. Not present if no calling party information is available.
<i>read</i>	EmptyContent	No	When present, indicates that the message has been read. When not present, indicates that the message has not previously been read.

Parameter	Parameter Type	Required	Comments
<i>urgent</i>	EmptyContent	No	When present, indicates that the voice message is marked urgent. When not present, the message is not urgent.
<i>confidential</i>	EmptyContent	No	When present, indicates that the voice message is marked confidential. When not present, the message is not confidential.
<i>video</i>	EmptyContent	No	When present, indicates that the voice mail is a video message. When not present, the message is voice only.
<i>fax</i>	EmptyContent	No	When present, indicates that the message is a fax message. When not present, the message is not a fax message.
<i>time</i>	Timestamp	Yes	The time the message was created.
<i>faxPageCount</i>	PositiveInt	No	The number of pages in this fax. Only present for fax messages.
<i>messageId</i>	String	No	The URI to access the message for downloading, marking as read/unread or deletion. This is not present for messages created prior to Release 20.0.

8.15.1.3 VoiceMailPartyInformation

A VoiceMailPartyInformation contains the general information for a party who left a voice mail. The following type definition table lists the content of a VoiceMailPartyInformation.

Parameter	Parameter Type	Required	Comments
<i>name</i>	NonEmptyString	No	The name of the party. Not present when there is no name or the user/application is not allowed to see it (the appropriate CLIP/COLP service is not enabled, or privacy is restricted and the application is not trusted). Not present for messages created prior to Release 20.0.
<i>userId</i>	UserId	No	The subscriber Id of the calling party. Only present when the party is in the same Group/Enterprise as the local user.
<i>address</i>	Address	No	The address of the calling party. Not present when there is no address (that is, the appropriate CLIP/COLP service is not enabled) or the user/application is not allowed to see it (that is, privacy is restricted and the application is not trusted, see the privacy parameter).
<i>privacy</i>	EmptyContent	No	When present, indicates that CLIP/COLP presentation is restricted. If present, then the address parameter is not present. If not present, then the address parameter may be present (if available).

8.15.2 VoiceMessage

A VoiceMessage contains an individual voice message. The following type definition table lists the content of a VoiceMessage.

Parameter	Parameter Type	Required	Comments
<i>messageInfo</i>	MessageInfo	Yes	
<i>messageMediaContent</i>	messageMediaContent	Yes	

8.15.2.1 MessageMediaContent

A MessageMediaContent represents the contents of a message to transfer with a description. The following type definition table lists the content of a MessageMediaContent.

Parameter	Parameter Type	Required	Comments
<i>description</i>	FileDescription	Yes	
<i>mediaType</i>	MessageMediaFileType	Yes	
<i>Content</i>	Base64Binary	Yes	

8.15.2.1.1 MessageMediaFileType

A MessageMediaFileType specified the media type of message data. The following table provides the list of values that can be assigned to a MessageMediaFileType.

File Type	Description
TIF	Tagged Image File Format (for fax messages).
WAV	A WAV file
MOV	A MOV file
MP3	An MP3 file

9 Message Flows

In this section, examples of actions, responses, and events are shown after the activity table. Each message is shown as the list of relevant parameters, the parameter types, and example values.

The following subscriber and their parameters are used for the call flows:

Identity	User Id	Name	DN/Extension
S1	subscriberS1@broadsoft.com	subscriberS1	Dn: 5006001011 Ext:1011
S2	subscriberS2@broadsoft.com	subscriberS2	Dn: 5006001012 Ext:1012
S3	subscriberS3@broadsoft.com	subscriberS3	Dn: 5006001013 Ext:1013
N1	Not applicable	Network	Dn: 5007003333
N2	Not applicable	Network2	Dn: 5007003334
ACD1	ACD_CTI1@broadsoft.com	ACDName	Dn: 5006001025 Ext: 1025
ACD2	ACD_CTI2@broadsoft.com	ACD2Name	Dn: 5006001027 Ext: 1027
RP	routePoint@broadsoft.com	RoutePointName	Dn: 5006001030 Ext: 1002
A1	agentA1@broadsoft.com	agentA1	Dn: 3004003000 Ext:3000

NOTE: Some of the call flows involve agents and supervisors. The agent functions are covered by a subscriber that has an agent Call Center license assigned (Basic, Standard, or Premium). The Supervisor functions are covered by a subscriber configured to supervise agents. A supervisor has agents assigned.

9.1 Call Message Flows

The following scenarios describe third-party call control situations where the remote application performs various requests on behalf of a subscriber. The same scenario could be reproduced in a first-party call control manner where a subscriber would initiate the same requests from the endpoint device. These scenario variants would produce the exact same events as the ones described for the third-party call control cases. Because the events are the same and the requests are equivalent, the first-party call control cases are not described in this document. In the end, they are just a subset of the described scenarios.

The next subsections illustrate the following scenario:

- Dial
- Hold
- Retrieve
- Release

- Consultative Call
- Call Forwarding
- Releasing
- Blind Transfer
- Consultative Transfer
- Mute Transfer
- Reconnect
- Transfer to Voice Mail
- Call Return (two-level mode)
- Directed Call Pickup
- Directed Call Pickup with Barge-in
- Monitor call

9.1.1 Dial

This scenario illustrates a successful Dial action from subscriber S1 to subscriber S2. In this scenario, both devices are available and valid, subscriber S1 is permitted to make the call and the call is answered by subscriber S2.

In this scenario, the subscriber S1 is prompted to go off-hook before subscriber S2 is called.

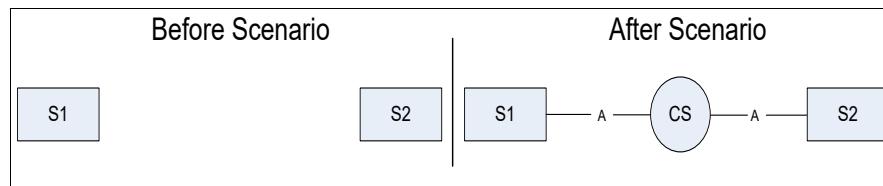


Figure 113 Dial Message Flow

Activity	Monitored Subscriber S1	Monitored Subscriber S2	Comments
A Dial action to a valid device is invoked on behalf of subscriber S1.	Request: Dial (F1) Dial Response (F2)		The Dial action specifies that subscriber S1 should be prompted to go off-hook.
Indication that the service has been initiated from this subscriber.	Call Originating Event (F3) Event Response (F4)		The "Click-to-Dial" personality indicates that the subscriber S1 is being prompted (via ringing, for example) to go off-hook.
Indication that the subscriber's endpoint supports the SIP talk event package.	Call Updated Event (F5) Event Response (F6)		
Subscriber S1 goes off-hook and is connected in the call.	Hook Status Event (F7) Event Response (F8) Call Originated Event(F9) Event Response (F10)		Subscriber S1 is now off-hook, the personality is 'Originator' and the call continues towards its destination.



Activity	Monitored Subscriber S1	Monitored Subscriber S2	Comments
Subscriber S2 begins to ring and S1 receives ringing tone.		Call Received Event(F11) Event Response (F12)	
Subscriber S2 answers the call by manually going off-hook.	Call Answered Event (F13) Event Response (F14))	Hook Status Event (F15) Event Response (F16) Call Answered Event (F17) Event Response (F18)	Active for both the users.

F1 Request

Parameter Name	Parameter Value
requestId	3
sessionId	MySessionId
credentials	YWRtaW5AdmlydHNhbml0eS5tdGwuYnJvYWRzb2Z0LmNvbTphZG1pbg==
DialRequest	Not applicable
Uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/new?%address%
method	POST
version	Not applicable
params	Not applicable
address	1012
subscriberId	subscriberS1@broadsoft.com
payload	Not applicable

F2 Response

Parameter Name	Parameter Value
RequestId	3
SessionId	168
statusCode	201
reason	Created
payload	Not applicable
CallStartInfo	Not applicable
callId	callhalf-10201:0
externalTrackingId	62:1

F3 CallOriginatingEvent

Parameter Name	Parameter Value
RequestId	0:1971
payload	Not applicable
Event	Not applicable
eventID	46228196-3dfa-45c3-a4d5-2e7e53c10d7e

Parameter Name	Parameter Value
sequenceNumber	8
userId	admin@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	ceaec03b-a6b7-4681-b4de-287d34cc2b89
channelId	a94890bf-73ae-43d4-87e4-5da75e9a28f7
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-10201:0
extTrackingId	62:1
personality	Click-to-Dial
state	Alerting
remoteParty	Not applicable
address	tel:1012
callType	Unknown
appearance	1
startTime	1271253522275

F4 EventResponse

Parameter Name	Parameter Value
RequestId	0:1971
Payload	Not applicable
EventResponse	Not applicable
Eventide	46228196-3dfa-45c3-a4d5-2e7e53c10d7e
statusCode	200
Reason	OK

F5 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:1972
payload	Not applicable
Event	Not applicable
eventID	7e81196c-2d33-4663-bce5-f3cea75a835a
sequenceNumber	9
userId	admin@broadsoft.com
externalApplicationId	AppCtlId

Parameter Name	Parameter Value
subscriptionId	ceaec03b-a6b7-4681-b4de-287d34cc2b89
channelId	a94890bf-73ae-43d4-87e4-5da75e9a28f7
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-10201:0
extTrackingId	62:1
personality	Click-to-Dial
state	Alerting
remoteParty	Not applicable
address	tel:1012
callType	Unknown
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
appearance	1
allowAnswer	Not applicable
startTime	1271253522275

F6 EventResponse

Parameter Name	Parameter Value
RequestId	0:1972
Payload	Not applicable
EventResponse	Not applicable
Eventide	7e81196c-2d33-4663-bce5-f3cea75a835a
StatusCode	200
Reason	OK

F7 HookStatusEvent

Parameter Name	Parameter Value
RequestId	0:1973
Payload	Not applicable
Event	Not applicable
EventID	68ea76c2-660e-43b1-b1ff-f32931db2854
sequenceNumber	10
UserId	admin@broadsoft.com
externalApplicationId	AppCtlId



Parameter Name	Parameter Value
subscriptionId	ceaec03b-a6b7-4681-b4de-287d34cc2b89
channelId	a94890bf-73ae-43d4-87e4-5da75e9a28f7
Targeted	subscriberS1@broadsoft.com
eventData	Not applicable
hookStatus	Off-Hook

F8 EventResponse

Parameter Name	Parameter Value
requestId	0:1973
Payload	Not applicable
EventResponse	Not applicable
eventide	68ea76c2-660e-43b1-b1ff-f32931db2854
statusCode	200
reason	OK

F9 CallOriginatedEvent

Parameter Name	Parameter Value
RequestId	0:1974
Payload	Not applicable
Event	Not applicable
Eventide	68ea76c2-660e-43b1-b1ff-f32931db2855
sequenceNumber	11
UserId	admin@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	ceaec03b-a6b7-4681-b4de-287d34cc2b89
channelId	a94890bf-73ae-43d4-87e4-5da75e9a28f7
Targeted	subscriberS1@broadsoft.com
eventData	Not applicable
Call	Not applicable
CallId	callhalf-10201:0
extTrackingId	62:1
personality	Originator
State	Alerting
remoteParty	Not applicable
Name	subscriberS2FirstName subscriberS2LastName
Address	tel:1012



Parameter Name	Parameter Value
UserId	subscriberS2@broadsoft.com
UserDN	tel:+15006001012;ext=1012
callType	Group
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
appearance	1
startTime	1271253524007

F10 EventResponse

Parameter Name	Parameter Value
requestId	0:1974
Payload	Not applicable
EventResponse	Not applicable
eventide	68ea76c2-660e-43b1-b1ff-f32931db2855
statusCode	200
reason	OK

F11 CallReceivedEvent

Parameter Name	Parameter Value
RequestId	0:1975
payload	Not applicable
Event	Not applicable
eventID	42b7594b-460e-4147-8a58-b7c8e5828fc5
sequenceNumber	12
userId	admin@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	ceaec03b-a6b7-4681-b4de-287d34cc2b89
channelId	a94890bf-73ae-43d4-87e4-5da75e9a28f7
targetId	subscriberS2@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-10205:0
extTrackingId	62:1
personality	Terminator
state	Alerting
remoteParty	Not applicable



Parameter Name	Parameter Value
name	subscriberS1FirstName subscriberS1LastName
address	tel:1011
userId	subscriberS1@broadsoft.com
userDN	tel:+15006001011;ext=1011
callType	Group
startTime	1271253524007

F12 EventResponse

Parameter Name	Parameter Value
RequestId	0:1975
Payload	Not applicable
EventResponse	Not applicable
Eventide	42b7594b-460e-4147-8a58-b7c8e5828fc5
StatusCode	200
Reason	OK

F13 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:1976
Payload	Not applicable
Event	Not applicable
eventID	1c66827b-650c-4737-8eab-8b579cd5e461
sequenceNumber	13
userId	admin@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	ceaec03b-a6b7-4681-b4de-287d34cc2b89
channelId	a94890bf-73ae-43d4-87e4-5da75e9a28f7
targetId	subscriberS2@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-10205:0
extTrackingId	62:1
personality	Terminator
state	Active
remoteParty	Not applicable
name	subscriberS1FirstName subscriberS1LastName



Parameter Name	Parameter Value
Address	tel:1011
userId	subscriberS1@broadsoft.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	Not applicable
addressOfRecord	5006001012@broadsoft.com
appearance	1
startTime	1271253524007
answerTime	1271253525554

F14 EventResponse

Parameter Name	Parameter Value
RequestId	0:1976
Payload	Not applicable
EventResponse	Not applicable
Eventide	1c66827b-650c-4737-8eab-8b579cd5e461
statusCode	200
Reason	OK

F15 HookStatusEvent

Parameter Name	Parameter Value
RequestId	0:1977
Payload	Not applicable
Event	Not applicable
EventID	68ea76c2-660e-43b1-b1ff-f32931db2856
sequenceNumber	10
UserId	admin@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	ceaec03b-a6b7-4681-b4de-287d34cc2b89
channelId	a94890bf-73ae-43d4-87e4-5da75e9a28f7
Targeted	subscriberS2@broadsoft.com
eventData	Not applicable
hookStatus	Off-Hook

F16 EventResponse

Parameter Name	Parameter Value
requestId	0:1973

Parameter Name	Parameter Value
Payload	Not applicable
EventResponse	Not applicable
eventide	68ea76c2-660e-43b1-b1ff-f32931db2856
statusCode	200
reason	OK

F17 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:1977
Payload	Not applicable
Event	Not applicable
eventide	4d287abc-db00-485d-bb21-3f9122b24796
sequenceNumber	14
userId	admin@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	ceaec03b-a6b7-4681-b4de-287d34cc2b89
channelId	a94890bf-73ae-43d4-87e4-5da75e9a28f7
targeted	subscriberS1@broadsoft.com
EventData	Not applicable
Call	Not applicable
callId	callhalf-10201:0
extTrackingId	62:1
personality	Originator
State	Active
remoteParty	Not applicable
Name	subscriberS2FirstName subscriberS2LastName
Address	tel:1012
userId	subscriberS2@broadsoft.com
userDN	tel:+15006001012;ext=1012
callType	Group
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
appearance	1
startTime	1271253524007
answerTime	1271253525554

F18 EventResponse

Parameter Name	Parameter Value
RequestId	0:1977
Payload	Not applicable
EventResponse	Not applicable
Eventide	4d287abc-db00-485d-bb21-3f9122b24796
StatusCode	200
Reason	OK

9.1.2 Hold

This scenario illustrates a successful Hold Request performed on call C1. In this scenario, both calls are in the *Active* state and subscriber S1's device accepts third-party call control hold request which places the active call to Subscriber S2 on hold.

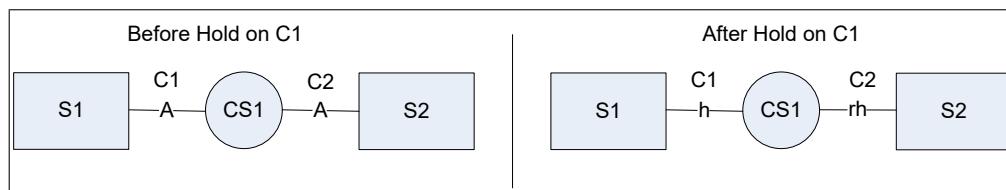


Figure 114 Hold Message Flow

Activity	Monitored Subscriber S1	Monitored Subscriber S2	Comments
A Hold request is issued on behalf of S1.	Hold Request (F1) Request Response (F2)		
Subscriber S1 call is placed on hold.	Call Held Event (F3) Event Response (F4)	Call Held Event (F5) Event Response (F6)	

F1 Request

Parameter Name	Parameter Value
RequestId	7
SessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
HoldRequest	Not applicable
Uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/%callId%/hold
Method	PUT
Version	Not applicable
Params	Not applicable
CallId	callhalf-11283:0
subscriberId	subscriberS1@broadsoft.com
Payload	Not applicable

F2 Response

Parameter Name	Parameter Value
RequestId	7
SessionId	228
StatusCode	200
Reason	OK
Payload	Not applicable

F3 CallHeldEvent

Parameter Name	Parameter Value
RequestId	0:2275
Payload	Not applicable
Event	Not applicable
Eventide	acfa58d1-a49a-4832-9fe4-60875fe996ba
sequenceNumber	12
UserId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	f650bd48-4785-4369-9f2d-16ba3ae6c0a3
channelId	681e0150-6bb5-4888-9758-3ffd056c0c1f
Targeted	subscriberS1@broadsoft.com
eventData	Not applicable
Call	Not applicable
CallId	callhalf-11283:0
extTrackingId	18:1
personality	Terminator
State	Held
remoteParty	Not applicable
Name	subscriberS2FirstName subscriberS2LastName
Address	tel:1012
UserId	subscriberS2@broadsoft.com
UserDN	tel:+15006001012;ext=1012
callType	Group
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
appearance	1
allowRetrieve	Not applicable



Parameter Name	Parameter Value
startTime	1271265744119
answerTime	1271265745924
heldTime	1271265909363

F4 EventResponse

Parameter Name	Parameter Value
RequestId	0:2275
Payload	Not applicable
EventResponse	Not applicable
Eventide	acfa58d1-a49a-4832-9fe4-60875fe996ba
StatusCode	200
Reason	OK

F5 CallHeldEvent

Parameter Name	Parameter Value
RequestId	0:2276
Payload	Not applicable
Event	Not applicable
Eventide	3f3499dd-a4d5-479e-a57e-9dec0fb500c3
sequenceNumber	13
UserId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	f650bd48-4785-4369-9f2d-16ba3ae6c0a3
Channeled	681e0150-6bb5-4888-9758-3ffd056c0c1f
Targeted	subscriberS2@broadsoft.com
EventData	Not applicable
Call	Not applicable
CallId	callhalf-11279:0
extTrackingId	18:1
Personality	Originator
State	Remote Held
remoteParty	Not applicable
Name	subscriberS1FirstName subscriberS1LastName
Address	tel:1011
UserId	subscriberS1@broadsoft.com
UserDN	tel:+15006001011;ext=1011

Parameter Name	Parameter Value
CallType	Group
Endpoint	Not applicable
addressOfRecord	5006001012@broadsoft.com
Appearance	1
StartTime	1271265744119
answerTime	1271265745924

F6 EventResponse

Parameter Name	Parameter Value
RequestId	0:2276
Payload	Not applicable
EventResponse	Not applicable
Eventide	3f3499dd-a4d5-479e-a57e-9dec0fb500c3
statusCode	200
Reason	OK

9.1.3 Retrieve

This scenario illustrates a successful Talk Request performed on call C1 while it is in the *Held* state. In this scenario subscriber S1's device accept third-party call control talk request to take the held call with Subscriber S2 and make it active.

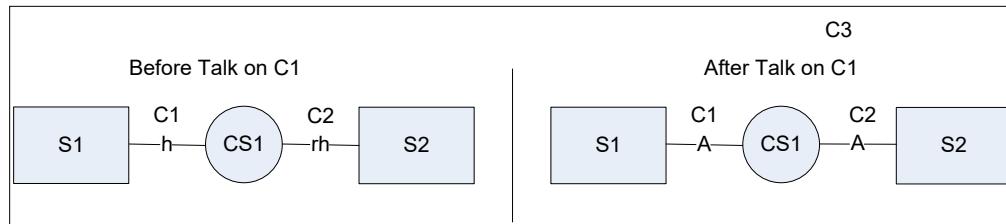


Figure 115 Retrieve Message Flow

Activity	Monitored Device S1	Monitored Device S2	Comments
The remote application issues a Talk request to retrieve the held call.	Talk Request (F1) Talk Response (F2)		
Both call C1 and C2 go back in Active state.	Call Retrieved Event (F3) Event Response (F4)	Call Retrieved Event (F5) Event Response (F6)	The Call Retrieved event specifies that call C1 and C2 are now back in the Active state.



F1 Request

Parameter Name	Parameter Value
RequestId	13
SessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
TalkRequest	Not applicable
Uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/%callId%/talk
Method	PUT
Version	Not applicable
Params	Not applicable
CallId	callhalf-11407:0
SubscriberId	subscriberS1@broadsoft.com
Payload	Not applicable

F2 Response

Parameter Name	Parameter Value
RequestId	13
SessionId	258
StatusCode	200
Reason	OK
payload	Not applicable

F3 CallRetrievedEvent

Parameter Name	Parameter Value
requestId	10:2344
payload	Not applicable
Event	Not applicable
eventID	d57dbfa5-7d37-496f-b7ad-d9855c43ce48
sequenceNumber	7
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	c47cd046-6b8f-48c1-be94-7ddb28a96aee
channelId	58131467-0a00-45c5-bdd9-544270013663
targetId	subscriberS1@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-11407:0



Parameter Name	Parameter Value
extTrackingId	26:1
personality	Terminator
state	Active
remoteParty	Not applicable
name	subscriberS2FirstName subscriberS2LastName
address	tel:1012
userId	subscriberS2@broadsoft.com
userDN	tel:+15006001012;ext=1012
callType	Group
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
appearance	1
startTime	1271267126217
answerTime	1271267126953
totalHeldTime	80336

F4 EventResponse

Parameter Name	Parameter Value
requestId	10:2344
payload	Not applicable
EventResponse	Not applicable
eventId	d57dbfa5-7d37-496f-b7ad-d9855c43ce48
statusCode	200
reason	OK

F5 CallRetrievedEvent

Parameter Name	Parameter Value
RequestId	10:2345
Payload	Not applicable
Event	Not applicable
Eventide	deca297a-5e42-4f58-992e-52e2bad56dbc
sequenceNumber	8
UserId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	c47cd046-6b8f-48c1-be94-7ddb28a96aee
channelId	58131467-0a00-45c5-bdd9-544270013663

Parameter Name	Parameter Value
Targeted	subscriberS2@broadsoft.com
eventData	Not applicable
Call	Not applicable
CallId	callhalf-11403:0
extTrackingId	26:1
personality	Originator
State	Active
remoteParty	Not applicable
Name	subscriberS1FirstName subscriberS1LastName
Address	tel:1011
UserId	subscriberS1@broadsoft.com
UserDN	tel:+15006001011;ext=1011
callType	Group
endpoint	Not applicable
addressOfRecord	5006001012@broadsoft.com
appearance	1
startTime	1271267126217
answerTime	1271267126953

F6 EventResponse

Parameter Name	Parameter Value
RequestId	10:2345
Payload	Not applicable
EventResponse	Not applicable
Eventide	deca297a-5e42-4f58-992e-52e2bad56dbc
StatusCode	200
Reason	OK

9.1.4 Release

This scenario illustrates a successful Release. Subscriber S1 and S2 calls are both in the Active state when the remote application initiates a release on call C1.

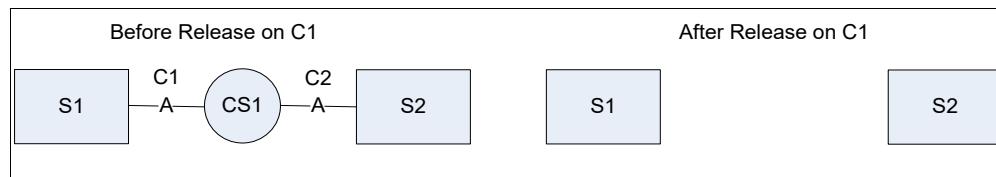


Figure 116 Release Message Flow

Activity	Monitored Device S1	Monitored Device S2	Comments
The remote application sends a Release request against call C1.	Release Request (F1) Release Response (F2)		
Since S2 is the only subscriber in the call, it is cleared as the result of S1 being cleared.	Call Released Event (F3) Event Response (F4) Hook Status Event (F5) Event Response (F6)	Call Released Event (F7) Event Response (F8) Hook Status Event (F9) Event Response (F10)	

F1 Request

Parameter Name	Parameter Value
RequestId	4
SessionId	MySessionId
credentials	YWRtaW5DVElAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
ReleaseRequest	Not applicable
Uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/%callId%
Method	DELETE
Version	Not applicable
Params	Not applicable
CallId	callhalf-11407:0
subscriberId	subscriberS1@broadsoft.com
Payload	Not applicable

F2 Response

Parameter Name	Parameter Value
RequestId	4
SessionId	280
statusCode	200
reason	OK
payload	Not applicable



F3 CallReleasedEvent

Parameter Name	Parameter Value
RequestId	0:2388
Payload	Not applicable
Event	Not applicable
eventID	79e9706f-676a-4ffd-b24b-dd1ab1f39c41
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	cfb45092-433c-49cf-852c-337ae8f2bcd
channelId	116f5d24-7756-4517-8276-c9a69a2cadbf
targetId	subscriberS2@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-11403:0
extTrackingId	26:1
personality	Originator
state	Released
releasingParty	remoteRelease
remoteParty	Not applicable
name	subscriberS1FirstName subscriberS1LastName
address	tel:1011
userId	subscriberS1@broadsoft.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	Not applicable
addressOfRecord	5006001012@broadsoft.com
startTime	1271267126217
answerTime	1271267126953
releaseTime	1271269018907

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:2388
payload	Not applicable
EventResponse	Not applicable
eventID	79e9706f-676a-4ffd-b24b-dd1ab1f39c41



Parameter Name	Parameter Value
statusCode	200
reason	OK

F5 HookStatusEvent

Parameter Name	Parameter Value
RequestId	0: 2389
Payload	Not applicable
Event	Not applicable
EventID	68ea76c2-660e-43b1-b1ff-f32931db2857
sequenceNumber	10
UserId	admin@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	cfb45092-433c-49cf-852c-337ae8f2bcd
channelId	116f5d24-7756-4517-8276-c9a69a2cadbf
Targeted	subscriberS1@broadsoft.com
eventData	Not applicable
hookStatus	Off-Hook

F6 EventResponse

Parameter Name	Parameter Value
requestId	0: 2389
Payload	Not applicable
EventResponse	Not applicable
eventid	68ea76c2-660e-43b1-b1ff-f32931db2857
statusCode	200
reason	OK

F7 CallReleasedEvent

Parameter Name	Parameter Value
requestId	0:2390
payload	Not applicable
Event	Not applicable
eventId	5543dc91-6e12-40f4-ab08-97f51514a7e2
sequenceNumber	1
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	cfb45092-433c-49cf-852c-337ae8f2bcd

Parameter Name	Parameter Value
channelId	116f5d24-7756-4517-8276-c9a69a2cadbf
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-11407:0
extTrackingId	26:1
personality	Terminator
state	Released
releasingParty	localRelease
remoteParty	Not applicable
name	subscriberS2FirstName subscriberS2LastName
address	tel:1012
userId	subscriberS2@broadsoft.com
UserDN	tel:+15006001012;ext=1012
callType	Group
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
startTime	1271267126217
answerTime	1271267126953
totalHeldTime	80336
releaseTime	1271269018906

F8 EventResponse

Parameter Name	Parameter Value
requestId	0:2390
Payload	Not applicable
EventResponse	Not applicable
eventId	5543dc91-6e12-40f4-ab08-97f51514a7e2
statusCode	200
reason	OK

F9 HookStatusEvent

Parameter Name	Parameter Value
RequestId	0: 2391
Payload	Not applicable
Event	Not applicable

Parameter Name	Parameter Value
EventId	68ea76c2-660e-43b1-b1ff-f32931db2852
sequenceNumber	2
UserId	admin@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	cfb45092-433c-49cf-852c-337ae8f2bcdb
channelId	116f5d24-7756-4517-8276-c9a69a2cadbf
Targeted	subscriberS2@broadsoft.com
eventData	Not applicable
hookStatus	Off-Hook

F10 EventResponse

Parameter Name	Parameter Value
requestId	0: 2391
Payload	Not applicable
EventResponse	Not applicable
eventid	68ea76c2-660e-43b1-b1ff-f32931db2852
statusCode	200
reason	OK

9.1.5 Consultative Call

The following example shows an example of consultative call by subscriber S1. In the example, subscriber S1 has an active call with S2 when he triggers a consultative call towards S3.

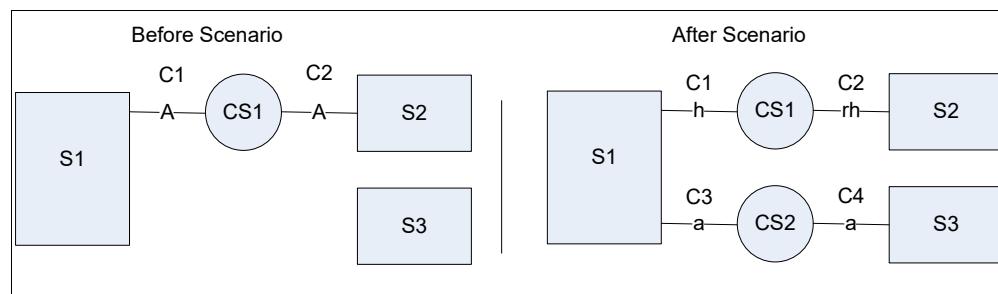


Figure 117 Consultative Call Message Flow

Activity	Monitored Subscriber S1	Monitored Subscriber S2	Monitored Subscriber S3	Comments
A Dial Request is performed on behalf of subscriber S1.	Dial Request (F1) Dial Response (F2)			Destination is S3



Activity	Monitored Subscriber S1	Monitored Subscriber S2	Monitored Subscriber S3	Comments
Indication that the service has been initiated from this subscriber.	Call Originating Event (F3) Event Response (F4)			The "Click-to-Dial" personality indicates that the subscriber S1 is being prompted (via ringing, for example) to go off-hook.
Indication that the subscriber endpoint supports the SIP talk event package.	Call Updated Event (F5) Event Response (F6)			
Subscriber S1 picks up the incoming call. C1 active call becomes <i>Held</i> .		Call Held Event (F7) Event Response (F8)		
C2 active call becomes <i>Remote Held</i> .	Call Held Event (F9) Event Response (F10)			
C3 is originated towards S3.	Call Originated Event (F13) Event Response (F14)			Call to S3 is in the <i>Alerting</i> state.
Indication that C4 has been received by S3.			Call Received Event (F11) Event Response (F12)	S3 has an incoming call from S1

F1 Request

Parameter Name	Parameter Value
requestId	14
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWZRzb2Z0LmNvbTpwYXNzd29yZA==
DialRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/new?%address%
method	POST
version	Not applicable
params	Not applicable
address	1013
subscriberId	subscriberS1@broadsoft.com
payload	Not applicable



F2 Response

Parameter Name	Parameter Value
requestId	14
sessionId	317
statusCode	201
reason	Created
payload	Not applicable
CallStartInfo	Not applicable
callId	callhalf-11927:3
externalTrackingId	108:1

F3 CallOriginatingEvent

Parameter Name	Parameter Value
requestId	11:2519
payload	Not applicable
Event	Not applicable
eventID	e0d6f292-c078-4bc5-bc51-b6ad6303db24
sequenceNumber	1
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	7f09c36a-3ee1-44f9-93fc-fe54fbf16519
channelId	4989a025-e801-4150-9bc1-8696c2e4d387
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-11927:3
extTrackingId	108:1
personality	Click-to-Dial
state	Alerting
RemoteParty	Not applicable
Address	tel:1013
CallType	Unknown
Appearance	2
StartTime	1271273069601



F4 EventResponse

Parameter Name	Parameter Value
RequestId	11:2519
Payload	Not applicable
EventResponse	Not applicable
EventId	e0d6f292-c078-4bc5-bc51-b6ad6303db24
StatusCode	200
Reason	OK

F5 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	11:2520
payload	Not applicable
Event	Not applicable
eventID	e5962816-28f6-466f-b39b-0df9c255d73d
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	7f09c36a-3ee1-44f9-93fc-fe54fbf16519
channelId	4989a025-e801-4150-9bc1-8696c2e4d387
targetId	subscriberS1@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-11927:3
extTrackingId	108:1
personality	Click-to-Dial
state	Alerting
remoteParty	Not applicable
address	tel:1013
callType	Unknown
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
appearance	2
allowAnswer	Not applicable
startTime	1271273069601



F6 EventResponse

Parameter Name	Parameter Value
requestId	11:2520
payload	Not applicable
EventResponse	Not applicable
eventID	e5962816-28f6-466f-b39b-0df9c255d73d
statusCode	200
reason	OK

F7 CallHeldEvent

Parameter Name	Parameter Value
requestId	11:2522
payload	Not applicable
Event	Not applicable
eventID	52d33d5d-33e7-46f3-bcff-46bd2d672278
sequenceNumber	4
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	7f09c36a-3ee1-44f9-93fc-fe54fbf16519
channelId	4989a025-e801-4150-9bc1-8696c2e4d387
targetId	subscriberS2@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-11923:0
extTrackingId	75:1
personality	Originator
state	Remote Held
remoteParty	Not applicable
name	subscriberS1FirstName subscriberS1LastName
address	tel:1011
userId	subscriberS1@broadsoft.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	Not applicable
addressOfRecord	5006001012@broadsoft.com
appearance	1
startTime	1271270976845



Parameter Name	Parameter Value
answerTime	1271270980007

F8 EventResponse

Parameter Name	Parameter Value
requestId	11:2522
payload	Not applicable
EventResponse	Not applicable
Eventide	52d33d5d-33e7-46f3-bcff-46bd2d672278
statusCode	200
Reason	OK

F9 CallHeldEvent

Parameter Name	Parameter Value
requestId	11:2521
Payload	Not applicable
Event	Not applicable
Eventide	b65927ec-62ef-43a5-b397-069f0bae81b1
sequenceNumber	3
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	7f09c36a-3ee1-44f9-93fc-fe54fbf16519
channelId	4989a025-e801-4150-9bc1-8696c2e4d387
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-11927:0
extTrackingId	75:1
personality	Terminator
state	Held
remoteParty	Not applicable
name	subscriberS2FirstName subscriberS2LastName
address	tel:1012
userId	subscriberS2@broadsoft.com
userDN	tel:+15006001012;ext=1012
callType	Group
endpoint	Not applicable



Parameter Name	Parameter Value
addressOfRecord	5006001011@broadsoft.com
appearance	1
allowRetrieve	Not applicable
startTime	1271270976845
answerTime	1271270980007
heldTime	1271273071553
totalHeldTime	2009900

F10 EventResponse

Parameter Name	Parameter Value
requestId	11:2521
payload	Not applicable
EventResponse	Not applicable
eventID	b65927ec-62ef-43a5-b397-069f0bae81b1
statusCode	200
reason	OK

F11 CallReceivedEvent

Parameter Name	Parameter Value
requestId	11:2524
payload	Not applicable
Event	Not applicable
eventID	eaf8103b-1e6f-4f72-b254-8a719ff3ea90
sequenceNumber	6
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	7f09c36a-3ee1-44f9-93fc-fe54fbf16519
channelId	4989a025-e801-4150-9bc1-8696c2e4d387
targetId	subscriberS3@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-12215:0
extTrackingId	108:1
personality	Terminator
state	Alerting
remoteParty	Not applicable



Parameter Name	Parameter Value
name	subscriberS1FirstName subscriberS1LastName
address	tel:1011
userId	subscriberS1@broadsoft.com
userDN	tel:+15006001011;ext=1011
callType	Group
startTime	1271273072015

F12 EventResponse

Parameter Name	Parameter Value
requestId	11:2524
payload	Not applicable
EventResponse	Not applicable
eventID	eaf8103b-1e6f-4f72-b254-8a719ff3ea90
statusCode	200
reason	OK

F13 CallOriginatedEvent

Parameter Name	Parameter Value
requestId	11:2523
payload	Not applicable
Event	Not applicable
eventID	302cb2c3-99bd-43a7-ba1b-1fef618e5e81
sequenceNumber	5
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	7f09c36a-3ee1-44f9-93fc-fe54fbf16519
channelId	4989a025-e801-4150-9bc1-8696c2e4d387
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-11927:3
extTrackingId	108:1
personality	Originator
state	Alerting
remoteParty	Not applicable
name	subscriberS3FirstName subscriberS3LastName



Parameter Name	Parameter Value
Address	tel:1013
UserId	subscriberS3@broadsoft.com
UserDN	tel:+15006001013;ext=1013
CallType	Group
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
appearance	2
startTime	1271273072015

F14 EventResponse

Parameter Name	Parameter Value
requestId	11:2523
payload	Not applicable
EventResponse	Not applicable
eventID	302cb2c3-99bd-43a7-ba1b-1fef618e5e81
statusCode	200
reason	OK

F15 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	11:2525
payload	Not applicable
Event	Not applicable
eventID	8414b50e-77a0-46ae-ad3d-0b2e4b7b0051
sequenceNumber	7
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	7f09c36a-3ee1-44f9-93fc-fe54fbf16519
channelId	4989a025-e801-4150-9bc1-8696c2e4d387
targetId	subscriberS3@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-12215:0
extTrackingId	108:1
personality	Terminator
state	Active



Parameter Name	Parameter Value
remoteParty	Not applicable
name	subscriberS1FirstName subscriberS1LastName
address	tel:1011
userId	subscriberS1@broadsoft.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	Not applicable
addressOfRecord	5006001013@broadsoft.com
appearance	1
startTime	1271273072015
answerTime	1271273073756

F16 EventResponse

Parameter Name	Parameter Value
requestId	11:2525
payload	Not applicable
EventResponse	Not applicable
eventID	8414b50e-77a0-46ae-ad3d-0b2e4b7b0051
statusCode	200
reason	OK

F17 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	11:2526
payload	Not applicable
Event	Not applicable
eventID	9982cffa-ce6b-4004-8bb3-74b2753ebc86
sequenceNumber	8
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	7f09c36a-3ee1-44f9-93fc-fe54fbf16519
channelId	4989a025-e801-4150-9bc1-8696c2e4d387
targetId	subscriberS1@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-11927:3



Parameter Name	Parameter Value
extTrackingId	108:1
personality	Originator
state	Active
remoteParty	Not applicable
name	subscriberS3FirstName subscriberS3LastName
address	tel:1013
userId	subscriberS3@broadsoft.com
userDN	tel:+15006001013;ext=1013
callType	Group
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
appearance	2
startTime	1271273072015
answerTime	1271273073756

F18 EventResponse

Parameter Name	Parameter Value
requestId	11:2526
payload	Not applicable
EventResponse	Not applicable
eventID	9982cffa-ce6b-4004-8bb3-74b2753ebc86
statusCode	200
reason	OK

9.1.6 Call Forwarding

This scenario illustrates a successful Call Forwarding. Subscriber S2 initiates a call to Subscriber S1 who has Call Forwarding Always enabled to Subscriber S3. In this scenario Subscriber S3 is available and valid, subscriber S1 is allowed to forward towards subscriber S3. Note that the example is partial, as it does not show S3 answering and releasing.

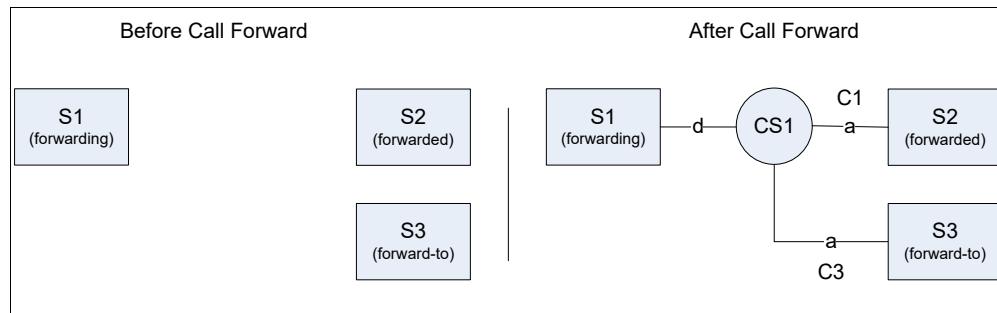


Figure 118 Call Forwarding Message Flow

Activity	Monitored Subscriber S1	Monitored Subscriber S2	Monitored Subscriber S3	Comments
S2 originates a call (C1) to S1.		Hook Status Event (F1) Event Response (F2) Call Originated Event (F3) Event Response (F4)		
Indication the subscriber S1 is receiving a call.	Call Received Event (F5) Event Response (F6)			
Subscriber S3 is alerted.			Call Received Event (F7) Event Response (F8)	The Call Received event contains detail on the redirection that was performed by S1
Indication that the call has been forwarded unconditionally.	Call Redirected Event (F9) Event Response (F10)	Call Forwarded Event (F11) Event Response (F12)		The Call Redirected event specifies that the redirection reason is Unconditional. S2 is notified that they are being forwarded.

F1 HookStatusEvent

Parameter Name	Parameter Value
requestId	0:1407
payload	N/A
Event	N/A
eventID	e9f3cf6d-a59e-4fff-a05e-90be7bf2875d
sequenceNumber	2
userId	asouth@broadworks.com



Parameter Name	Parameter Value
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	e4f29870-3372-4539-b8e5-be445f044009
channelId	7aa69821-be7f-4f84-9976-32aac4c72d0b
targetId	subscriberS2@broadworks.com
eventData	N/A
hookStatus	Off-Hook

F2 EventResponse

Parameter Name	Parameter Value
requestId	0:1407
payload	N/A
EventResponse	N/A
eventID	e9f3cf6d-a59e-4fff-a05e-90be7bf2875d
statusCode	200
reason	OK

F3 CallOriginatedEvent

Parameter Name	Parameter Value
requestId	0:1409
payload	N/A
Event	N/A
eventID	79746152-f9c8-4427-88d1-cea711e4cda9
sequenceNumber	3
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	e4f29870-3372-4539-b8e5-be445f044009
channelId	7aa69821-be7f-4f84-9976-32aac4c72d0b
targetId	subscriberS2@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-5139:0
extTrackingId	7cd733af-6a9f-4c8a-b213-54fe71d2d578
personality	Originator
state	Alerting
remoteParty	N/A
name	subscriberS1Firstname subscriberS1Lastname



Parameter Name	Parameter Value
address	tel:1011
userId	subscriberS1@broadworks.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	N/A
addressOfRecord	5006001012@broadworks.com
appearance	1
startTime	1382030177053

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:1409
payload	N/A
EventResponse	N/A
eventID	79746152-f9c8-4427-88d1-cea711e4cda9
statusCode	200
reason	OK

F5 CallReceivedEvent

Parameter Name	Parameter Value
requestId	0:1408
payload	N/A
Event	N/A
eventID	679e4a63-3cb1-423c-9912-0d97068a6887
sequenceNumber	2
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	44f35f0c-93f6-46c4-8ac9-dc2f76afe065
channelId	7aa69821-be7f-4f84-9976-32aac4c72d0b
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-5143:0
extTrackingId	7cd733af-6a9f-4c8a-b213-54fe71d2d578
personality	Terminator
state	Alerting



Parameter Name	Parameter Value
remoteParty	N/A
name	subscriberS2Firstname subscriberS2Lastname
address	tel:1012
userId	subscriberS2@broadworks.com
userDN	tel:+15006001012;ext=1012
callType	Group
startTime	1382030177053

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:1408
payload	N/A
EventResponse	N/A
eventID	679e4a63-3cb1-423c-9912-0d97068a6887
statusCode	200
reason	OK

F7 CallReceivedEvent

Parameter Name	Parameter Value
requestId	0:1412
payload	N/A
Event	N/A
eventID	6ec21323-9f97-44d4-85f9-259967f5cc44
sequenceNumber	2
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	4c3216d3-44c5-4437-89be-cd8042b65bb3
channelId	7aa69821-be7f-4f84-9976-32aac4c72d0b
targetId	subscribers3@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-5147:0
extTrackingId	7cd733af-6a9f-4c8a-b213-54fe71d2d578
personality	Terminator
state	Alerting
remoteParty	N/A



Parameter Name	Parameter Value
name	subscriberS2Firstname subscriberS2Lastname
address	tel:1012
userId	subscriberS2@broadworks.com
userDN	tel:+15006001012;ext=1012
callType	Group
redirections	N/A
redirection	N/A
party	N/A
name	subscriberS1Firstname subscriberS1Lastname
address	tel:+15006001011
userId	subscriberS1@broadworks.com
callType	Group
reason	unconditional
startTime	1382030177480

F8 EventResponse

Parameter Name	Parameter Value
requestId	0:1412
payload	N/A
EventResponse	N/A
eventID	6ec21323-9f97-44d4-85f9-259967f5cc44
statusCode	200
reason	OK

F9 CallRedirectedEvent

Parameter Name	Parameter Value
requestId	0:1410
payload	N/A
Event	N/A
eventID	fe5cb8d9-debe-44bd-8e0c-6c21081b8897
sequenceNumber	3
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	44f35f0c-93f6-46c4-8ac9-dc2f76afe065
channelId	7aa69821-be7f-4f84-9976-32aac4c72d0b
targetId	subscriberS1@broadworks.com

Parameter Name	Parameter Value
eventData	N/A
calls	N/A
call	N/A
callId	callhalf-5143:0
extTrackingId	7cd733af-6a9f-4c8a-b213-54fe71d2d578
personality	Terminator
state	Detached
remoteParty	N/A
name	subscriberS2Firstname subscriberS2Lastname
address	tel:1012
userId	subscriberS2@broadworks.com
userDN	tel:+15006001012;ext=1012
callType	Group
redirect	N/A
address	tel:1013
reason	unconditional
redirectTime	1382030177480
startTime	1382030177053
detachedTime	1382030177709

F10 EventResponse

Parameter Name	Parameter Value
requestId	0:1410
payload	N/A
EventResponse	N/A
eventID	fe5cb8d9-debe-44bd-8e0c-6c21081b8897
statusCode	200
reason	OK

F11 CallForwardedEvent

Parameter Name	Parameter Value
requestId	0:1411
payload	N/A
Event	N/A
eventID	22fc07ae-222a-42b3-9412-96124e1e8479
sequenceNumber	4



Parameter Name	Parameter Value
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	e4f29870-3372-4539-b8e5-be445f044009
channelId	7aa69821-be7f-4f84-9976-32aac4c72d0b
targetId	subscriberS2@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-5139:0
extTrackingId	7cd733af-6a9f-4c8a-b213-54fe71d2d578
personality	Originator
state	Alerting
remoteParty	N/A
name	subscribers3Firstname subscribers3Lastname
address	tel:1013
userId	subscribers3@broadworks.com
userDN	tel:+15006001013;ext=1013
callType	Group
endpoint	N/A
addressOfRecord	5006001012@broadworks.com
appearance	1
startTime	1382030177053

F12 EventResponse

Parameter Name	Parameter Value
requestId	0:1411
payload	N/A
EventResponse	N/A
eventId	22fc07ae-222a-42b3-9412-96124e1e8479
statusCode	200
reason	OK

9.1.7 Releasing

This scenario illustrates the releasing state when call treatment is offered to the originator. Subscriber S1 tries initiating a call to a non-existent location due to a misdial. As such, S1 receives a treatment. While the treatment is being played, the monitored subscriber receives a Releasing event with the release reason set.

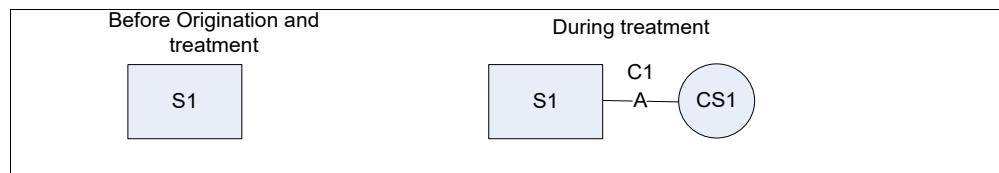


Figure 119 Releasing Message Flow

Activity	Monitored Device S1	Comments
Subscriber S1 originates a call to a non-existent location.	Hook Status Event (F1) Event Response (F2) Call Originated Event (F3) Event Response (F4)	Call dialed using S1's device.
Subscriber S1 receives a treatment.	Call Releasing Event (F5) Event Response (F6)	Treatment is being played to the originator. Released Event will be sent when S1 hangs up.

F1 HookStatusEvent

Parameter Name	Parameter Value
requestId	0:1498
payload	N/A
Event	N/A
eventID	1cbf22d2-26e7-472d-81fe-ce0ec04e8f2c
sequenceNumber	2
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	930d5030-4a03-4b26-a898-f176ff9844af
channelId	71e4df15-45da-4b5b-b4b8-ce49ab80d63e
targetId	subscriberS1@broadworks.com
eventData	N/A
hookStatus	Off-Hook

F2 EventResponse

Parameter Name	Parameter Value
requestId	0:1498
payload	N/A
EventResponse	N/A
eventID	1cbf22d2-26e7-472d-81fe-ce0ec04e8f2c

Parameter Name	Parameter Value
statusCode	200
reason	OK

F3 CallOriginatedEvent

Parameter Name	Parameter Value
requestId	0:1499
payload	N/A
Event	N/A
eventID	ba098590-3c4b-4e95-b301-66458c77aa5c
sequenceNumber	3
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	930d5030-4a03-4b26-a898-f176ff984af
channelId	71e4df15-45da-4b5b-b4b8-ce49ab80d63e
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-5323:0
extTrackingId	16d236ff-2938-49ab-a2f9-647f7ef88f5a
personality	Originator
state	Active
releaseCause	N/A
internalReleaseCause	User Not Found
cdrTerminationCause	111
remoteParty	N/A
address	tel:10
callType	Network
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
startTime	1382032356918

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:1499
payload	N/A



Parameter Name	Parameter Value
EventResponse	N/A
eventID	ba098590-3c4b-4e95-b301-66458c77aa5c
statusCode	200
reason	OK

F5 CallReleasingEvent

Parameter Name	Parameter Value
requestId	0:1500
payload	N/A
Event	N/A
eventID	5e2a6773-ecad-4895-8bd2-b50d6a56f445
sequenceNumber	4
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	930d5030-4a03-4b26-a898-f176ff9844af
channelId	71e4df15-45da-4b5b-b4b8-ce49ab80d63e
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-5323:0
extTrackingId	16d236ff-2938-49ab-a2f9-647f7ef88f5a
personality	Originator
state	Active
releaseCause	N/A
internalReleaseCause	User Not Found
cdrTerminationCause	111
remoteParty	N/A
address	tel:10
callType	Network
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
startTime	1382032356918

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:1500
payload	N/A
EventResponse	N/A
eventID	5e2a6773-ecad-4895-8bd2-b50d6a56f445
statusCode	200
reason	OK

9.1.8 Blind Transfer

This scenario illustrates a successful Blind Transfer. Subscriber S1 and S2 calls are both in the *Active* state when the remote application initiates a blind transfer to subscriber S3. In this scenario, subscriber S3 is available and valid, subscriber S1 is permitted to make the transfer, and the call is answered by subscriber S3.

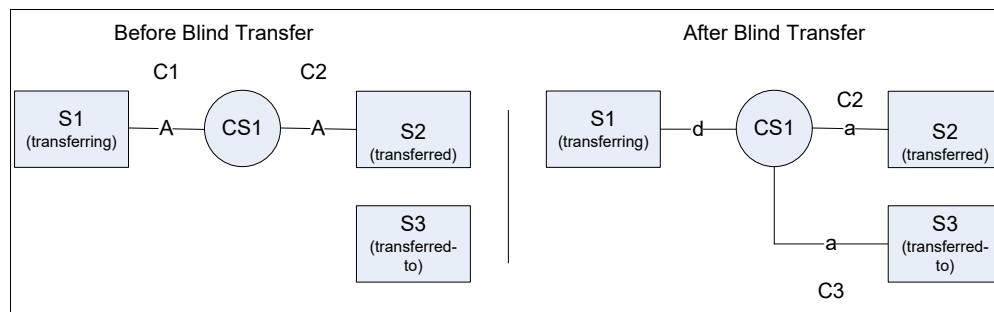


Figure 120 Blind Transfer Message Flow

Activity	Monitored Subscriber S1	Monitored Subscriber S2	Monitored Subscriber S3	Comments
A Transfer Request to subscriber S3 is invoked on behalf of subscriber S1.	Blind Transfer Request (F1) Blind Transfer Response (F2)			
Indication that the Request has been initiated from this subscriber.	Call Redirected Event (F3) Event Response (F4)	Call Transferred Event (F5) Event Response (F6)		The Call Transferred event specifies that the call is now in <i>Alerting</i> state and remoteParty is now S3.
Subscriber S3 is alerted.	Hook Status Event (F7) Event Response (F8)		Call Received Event (F9) Event Response (F10)	The Call Received event contains detail on the redirection that was performed.



F1 Request

Parameter Name	Parameter Value
requestId	5
sessionId	10
credentials	YXNvdXRoQG10bGFzZGV2ODcubmV0Om10bGxhYg==
BlindTransferRequest	N/A
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/%callId%/blindtransfer?%address%
method	PUT
version	N/A
params	N/A
address	tel:1013
callId	callhalf-5379:0
subscriberId	subscriberS1@broadworks.com
payload	N/A

F2 Response

Parameter Name	Parameter Value
requestId	5
sessionId	10
statusCode	200
reason	OK
payload	N/A

F3 CallRedirectedEvent

Parameter Name	Parameter Value
requestId	0:1538
payload	N/A
Event	N/A
eventID	54e72ae5-caae-47c3-a970-cdbbd6ad1562
sequenceNumber	11
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	930d5030-4a03-4b26-a898-f176ff9844af
channelId	71e4df15-45da-4b5b-b4b8-ce49ab80d63e
targetId	subscriberS1@broadworks.com
eventData	N/A
calls	N/A

Parameter Name	Parameter Value
call	N/A
callId	callhalf-5379:0
extTrackingId	238227c6-ee1d-4ceb-b3f1-984eec2ab1b9
personality	Terminator
state	Detached
remoteParty	N/A
name	subscriberS2Firstname subscriberS2Lastname
address	tel:1012
userId	subscriberS2@broadworks.com
userDN	tel:+15006001012;ext=1012
callType	Group
redirect	N/A
address	tel:1013
reason	transfer
redirectTime	1382033204629
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
startTime	1382033100556
answerTime	1382033102130
detachedTime	1382033204630

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:1538
payload	N/A
EventResponse	N/A
eventID	54e72ae5-caae-47c3-a970-cdbbd6ad1562
statusCode	200
reason	OK

F5 CallTransferredEvent

Parameter Name	Parameter Value
requestId	0:1539
payload	N/A
Event	N/A
eventID	5982bf51-f9c0-441e-88cd-625b7ead6cbb



Parameter Name	Parameter Value
sequenceNumber	5
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	dbb4f6af-6c40-4fb1-8726-dad76c4b2a16
channelId	71e4df15-45da-4b5b-b4b8-ce49ab80d63e
targetId	subscriberS2@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-5375:0
extTrackingId	238227c6-ee1d-4ceb-b3f1-984eec2ab1b9
personality	Originator
state	Alerting
remoteParty	N/A
name	subscribers3Firstname subscribers3Lastname
address	tel:1013
userId	subscribers3@broadworks.com
userDN	tel:+15006001013;ext=1013
callType	Group
endpoint	N/A
addressOfRecord	5006001012@broadworks.com
appearance	1
startTime	1382033100556
answerTime	1382033102130

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:1539
payload	N/A
EventResponse	N/A
eventId	5982bf51-f9c0-441e-88cd-625b7ead6cbb
statusCode	200
reason	OK



F7 HookStatusEvent

Parameter Name	Parameter Value
requestId	0:1540
payload	N/A
Event	N/A
eventID	428dbc7-01f9-464e-8dbf-fc29191f53c6
sequenceNumber	12
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	930d5030-4a03-4b26-a898-f176ff9844af
channelId	71e4df15-45da-4b5b-b4b8-ce49ab80d63e
targetId	subscriberS1@broadworks.com
eventData	N/A
hookStatus	On-Hook

F8 EventResponse

Parameter Name	Parameter Value
requestId	0:1540
payload	N/A
EventResponse	N/A
eventID	428dbc7-01f9-464e-8dbf-fc29191f53c6
statusCode	200
reason	OK

F9 CallReceivedEvent

Parameter Name	Parameter Value
requestId	0:1541
payload	N/A
Event	N/A
eventID	706852e9-989d-4cbc-9666-96c1cee0f140
sequenceNumber	2
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	875eacc5-7f14-4ed5-a2d4-9001ff28b864
channelId	71e4df15-45da-4b5b-b4b8-ce49ab80d63e
targetId	subscribers3@broadworks.com
eventData	N/A

Parameter Name	Parameter Value
call	N/A
callId	callhalf-5391:0
extTrackingId	238227c6-ee1d-4ceb-b3f1-984eec2ab1b9
personality	Terminator
state	Alerting
remoteParty	N/A
name	subscriberS2Firstname subscriberS2Lastname
address	tel:1012
userId	subscriberS2@broadworks.com
userDN	tel:+15006001012;ext=1012
callType	Group
redirections	N/A
redirection	N/A
party	N/A
name	subscriberS1Firstname subscriberS1Lastname
address	tel:+15006001011
userId	subscriberS1@broadworks.com
callType	Group
reason	deflection
startTime	1382033204629

F10 EventResponse

Parameter Name	Parameter Value
requestId	0:1541
payload	N/A
EventResponse	N/A
eventID	706852e9-989d-4cbc-9666-96c1cee0f140
statusCode	200
reason	OK

9.1.9 Consultative Transfer

This scenario illustrates a successful Consultative Transfer. Call C2 is in the *Held* state while call C3 is in the *Active* state when the remote application initiates a consultative transfer. In this scenario, subscriber S2 is permitted to make the transfer.

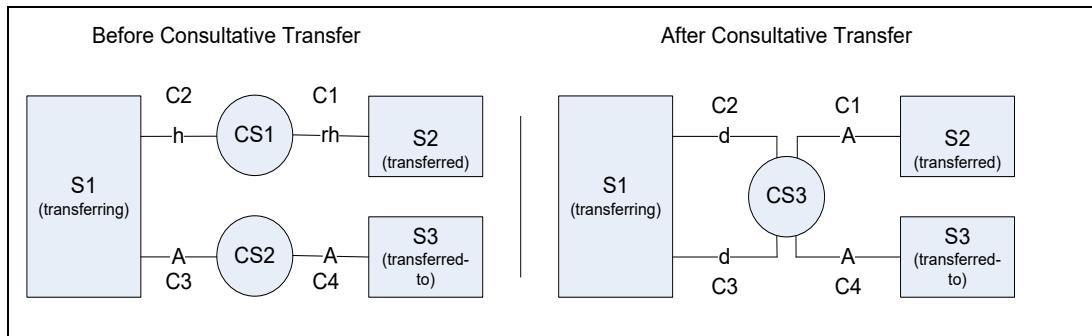


Figure 121 Consultative Transfer Message Flow

Activity	Monitored Subscriber S1	Monitored Subscriber S2	Monitored Subscriber S3	Comments
A Consultative Transfer Request is invoked on behalf of subscriber S1.	Request: Consultative Transfer (F1) Response (F6)			
Indication that the Request has been initiated from this subscriber.	Call Redirected Event (F7) Response (F8)	Call Transferred Event (F2) Response (F3)	Call Transferred Event (F4) Response (F5)	The Call Transferred events specify that the calls are now in state <i>Active</i> and remoteParty is set accordingly.
Indication that C2 is no longer in the <i>Held</i> state.		Call Retrieved Event (F9) Response (F10)		
Indication that S1 is on-hook.	Hook Status Event (F11) Response (F12)			

F1 Request

Parameter Name	Parameter Value
requestId	37
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
ConsultativeTransfer Request	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/%callId1%/consulttransfer/%callId2%
method	PUT

Parameter Name	Parameter Value
version	Not applicable
params	Not applicable
callId1	callhalf-63883:0
callId2	callhalf-63883:1
subscriberId	subscriberS1@broadsoft.com
payload	Not applicable

F2 CallTransferredEvent

Parameter Name	Parameter Value
requestId	32:569
payload	Not applicable
Event	Not applicable
eventID	2f4ddc49-ebf6-4abc-81d6-f35f5728cbf3
sequenceNumber	5
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	c13c9c7b-de6f-4e30-8fc4-784ecb74eca4
channelId	f63b07e0-445f-4bca-8311-ab305cf5402e
targetId	subscriberS2@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-63879:0
extTrackingId	536:1
personality	Terminator
state	Remote Held
remoteParty	Not applicable
name	subscriberS3FirstName subscriberS3LastName
address	tel:1013
userId	subscriberS3@broadsoft.com
userDN	tel:+15006001013;ext=1013
callType	Group
endpoint	Not applicable
addressOfRecord	5006001012@broadsoft.com
appearance	1
startTime	1271798055993
answerTime	1271798058313



F3 EventResponse

Parameter Name	Parameter Value
requestId	32:569
payload	Not applicable
EventResponse	Not applicable
eventID	2f4ddc49-ebf6-4abc-81d6-f35f5728cbf3
statusCode	200
reason	OK

F4 CallTransferredEvent

Parameter Name	Parameter Value
requestId	32:570
payload	Not applicable
Event	Not applicable
eventID	2f797c2c-1e7c-4e65-be13-307e2fcf8690
sequenceNumber	4
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	6e82e04f-abc2-487a-b3fc-7200f2515473
channelId	f63b07e0-445f-4bca-8311-ab305cf5402e
targetId	subscriberS3@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-63899:0
extTrackingId	536:1
personality	Originator
state	Active
remoteParty	Not applicable
name	subscriberS2FirstName subscriberS2LastName
address	tel:1012
userId	subscriberS2@broadsoft.com
userDN	tel:+15006001012;ext=1012
callType	Group
endpoint	Not applicable
addressOfRecord	5006001013@broadsoft.com
appearance	1



Parameter Name	Parameter Value
startTime	1271798112635
answerTime	1271798114251

F5 EventResponse

Parameter Name	Parameter Value
requestId	32:570
payload	Not applicable
EventResponse	Not applicable
eventID	2f797c2c-1e7c-4e65-be13-307e2fcf8690
statusCode	200
reason	OK

F6 Response

Parameter Name	Parameter Value
requestId	37
sessionId	270
statusCode	200
reason	Not applicable
payload	Not applicable

F7 CallRedirectedEvent

Parameter Name	Parameter Value
requestId	32:571
payload	Not applicable
Event	Not applicable
eventID	2ac1819a-de17-486d-8b06-dd0422c7359f
sequenceNumber	9
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	b3240ceb-9a4e-4fef-9e30-848817930dd3
channelId	f63b07e0-445f-4bca-8311-ab305cf5402e
targetId	subscriberS1@broadsoft.com
EventData	Not applicable
calls	Not applicable
call	Not applicable
callId	callhalf-63883:0
extTrackingId	536:1



Parameter Name	Parameter Value
personality	Terminator
state	Held
remoteParty	Not applicable
name	subscriberS2FirstName subscriberS2LastName
address	tel:1012
userId	subscriberS2@broadsoft.com
userDN	tel:+15006001012;ext=1012
callType	Group
redirect	Not applicable
address	tel:1013
reason	transfer
redirectTime	1271798576165
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
appearance	1
allowRetrieve	Not applicable
startTime	1271798055993
answerTime	1271798058313
heldTime	1271798113802
call	Not applicable
callId	callhalf-63883:1
extTrackingId	536:1
personality	Terminator
state	Detached
remoteParty	Not applicable
name	subscriberS3FirstName subscriberS3LastName
address	tel:1013
userId	subscriberS3@broadsoft.com
userDN	tel:+15006001013;ext=1013
callType	Group
redirect	Not applicable
address	tel:1012
reason	transfer
redirectTime	1271798576165
endpoint	Not applicable



Parameter Name	Parameter Value
addressOfRecord	5006001011@broadsoft.com
startTime	1271798112635
answerTime	1271798114251
detachedTime	1271798576166

F8 EventResponse

Parameter Name	Parameter Value
requestId	32:571
payload	Not applicable
EventResponse	Not applicable
eventID	2ac1819a-de17-486d-8b06-dd0422c7359f
statusCode	200
reason	OK

F9 CallRetrievedEvent

Parameter Name	Parameter Value
requestId	32:572
payload	Not applicable
Event	Not applicable
eventID	ffe89c56-57f3-445b-aa4c-9cb6c558de57
sequenceNumber	6
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	c13c9c7b-de6f-4e30-8fc4-784ecb74eca4
channelId	f63b07e0-445f-4bca-8311-ab305cf5402e
targetId	subscriberS2@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-63879:0
extTrackingId	536:1
personality	Terminator
state	Active
remoteParty	Not applicable
name	subscriberS3FirstName subscriberS3LastName
address	tel:1013
userId	subscriberS3@broadsoft.com



Parameter Name	Parameter Value
userDN	tel:+15006001013;ext=1013
callType	Group
endpoint	Not applicable
addressOfRecord	5006001012@broadsoft.com
appearance	1
startTime	1271798055993
answerTime	1271798058313

F10 EventResponse

Parameter Name	Parameter Value
requestId	32:572
payload	Not applicable
EventResponse	Not applicable
eventID	ffe89c56-57f3-445b-aa4c-9cb6c558de57
statusCode	200
reason	OK

F11 HookStatusEvent

Parameter Name	Parameter Value
requestId	32:573
payload	N/A
Event	N/A
eventID	428dbc7-01f9-464e-8dbf-fc29191f53d6
sequenceNumber	10
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	b3240ceb-9a4e-4fef-9e30-848817930dd3
channelId	f63b07e0-445f-4bca-8311-ab305cf5402e
targetId	subscriberS1@broadworks.com
EventData	N/A
hookStatus	On-Hook

F12 EventResponse

Parameter Name	Parameter Value
requestId	32:573
payload	N/A
EventResponse	N/A

Parameter Name	Parameter Value
eventID	428dbcb7-01f9-464e-8dbf-fc29191f53d6
statusCode	200
reason	OK

9.1.10 Mute Transfer

This scenario illustrates a successful Mute Transfer. Subscriber S1 and S2 calls are both in the *Active* state when the remote application initiates a mute transfer to subscriber S3. In this scenario, subscriber S3 is available and valid, subscriber S1 is permitted to make the transfer, and the call is answered by subscriber S3.

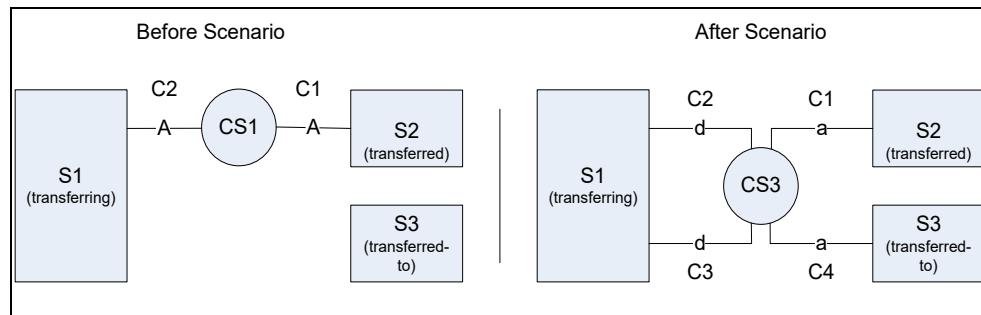


Figure 122 Mute Transfer Message Flow

Activity	Monitored Subscriber S1	Monitored Subscriber S2	Monitored Subscriber S3	Comments
A Mute Transfer Request to subscriber S3 is invoked on behalf of subscriber S1.	Mute Transfer Request (F1) Mute Transfer Response (F2)			
Indication that a new call is initiated toward subscriber S3.	Call Originated Event (F3) Event Response (F4)		Call Received Event (F5) Event Response (F6)	Subscriber S3 phone starts to ring.
Indication that the Mute Transfer is successful.	Call Redirected Event (F11) Event Response (F12) Hook Status Event (F13) Event Response (F14)	Call Transferred Event (F9) Event Response (F10)	Call Transferred Event (F7) Event Response (F8)	The Call Transferred event specifies that the call is now in <i>Alerting</i> state and remoteParty is now S3.

F1 Request

Parameter Name	Parameter Value
requestId	6
sessionId	15
credentials	YXNvdXRoQG10bGFzZGV2ODcubmV0Om10bGxhYg==



Parameter Name	Parameter Value
MuteTransferRequest	N/A
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/%callId%/mutetransfer?%address%
method	PUT
version	N/A
params	N/A
address	tel:1013
callId	callhalf-5627:0
subscriberId	subscriberS1@broadworks.com
payload	N/A

F2 Response

Parameter Name	Parameter Value
requestId	6
sessionId	15
statusCode	200
reason	OK
payload	N/A
CallStartInfo	N/A
callId	callhalf-5627:1
externalTrackingId	9a336cbb-8284-49af-a19b-06c755ad2bff

F3 CallOriginatedEvent

Parameter Name	Parameter Value
requestId	0:1681
payload	N/A
Event	N/A
eventID	1aff9c95-ab6c-4dfa-ae43-58fc72739277
sequenceNumber	10
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	48d32994-4030-415b-ac19-53d93ff36245
channelId	77f0f20c-f278-48ff-b409-7ca31ed522dc
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-5627:1



Parameter Name	Parameter Value
extTrackingId	9a336cbb-8284-49af-a19b-06c755ad2bff
personality	Originator
state	Alerting
remoteParty	N/A
name	subscribers3Firstname subscribers3Lastname
address	tel:1013
userId	subscribers3@broadworks.com
userDN	tel:+15006001013;ext=1013
callType	Group
startTime	1382035884272

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:1681
payload	N/A
EventResponse	N/A
eventID	1aff9c95-ab6c-4dfa-ae43-58fc72739277
statusCode	200
reason	OK

F5 CallReceivedEvent

Parameter Name	Parameter Value
requestId	0:1682
payload	N/A
Event	N/A
eventID	c96202b2-979f-4b64-ab01-7ac1737471c1
sequenceNumber	2
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d145fefc-bc60-474a-867c-879a5a960b97
channelId	77f0f20c-f278-48ff-b409-7ca31ed522dc
targetId	subscribers3@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-5635:0
extTrackingId	9a336cbb-8284-49af-a19b-06c755ad2bff



Parameter Name	Parameter Value
personality	Terminator
state	Alerting
remoteParty	N/A
name	subscriberS1Firstname subscriberS1Lastname
address	tel:1011
userId	subscriberS1@broadworks.com
userDN	tel:+15006001011;ext=1011
callType	Group
startTime	1382035884272

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:1682
payload	N/A
EventResponse	N/A
eventID	c96202b2-979f-4b64-ab01-7ac1737471c1
statusCode	200
reason	OK

F7 CallTransferredEvent

Parameter Name	Parameter Value
requestId	0:1683
payload	N/A
Event	N/A
eventID	f0c23151-be98-4485-b9a7-a8b57f05c899
sequenceNumber	3
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d145fefc-bc60-474a-867c-879a5a960b97
channelId	77f0f20c-f278-48ff-b409-7ca31ed522dc
targetId	subscribers3@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-5635:0
extTrackingId	14060905-2d7a-469e-a4e9-8c4d4ed823b0
personality	Terminator

Parameter Name	Parameter Value
state	Alerting
remoteParty	N/A
name	subscriberS2Firstname subscriberS2Lastname
address	tel:1012
userId	subscriberS2@broadworks.com
userDN	tel:+15006001012;ext=1012
callType	Group
endpoint	N/A
addressOfRecord	5006001013@broadworks.com
appearance	1
startTime	1382035884272

F8 EventResponse

Parameter Name	Parameter Value
requestId	0:1683
payload	N/A
EventResponse	N/A
eventID	f0c23151-be98-4485-b9a7-a8b57f05c899
statusCode	200
reason	OK

F9 CallTransferredEvent

Parameter Name	Parameter Value
requestId	0:1684
payload	N/A
Event	N/A
eventID	17d5c11c-1484-4f9a-8838-d4ab78b2332b
sequenceNumber	7
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	a593ceaf-ac3f-4c55-a6f2-6219a576ce16
channelId	77f0f20c-f278-48ff-b409-7ca31ed522dc
targetId	subscriberS2@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-5623:0



Parameter Name	Parameter Value
extTrackingId	14060905-2d7a-469e-a4e9-8c4d4ed823b0
personality	Originator
state	Alerting
remoteParty	N/A
name	subscribers3Firstname subscribers3Lastname
address	tel:1013
userId	subscribers3@broadworks.com
userDN	tel:+15006001013;ext=1013
callType	Group
endpoint	N/A
addressOfRecord	5006001012@broadworks.com
appearance	1
startTime	1382035846972
answerTime	1382035848055

F10 EventResponse

Parameter Name	Parameter Value
requestId	0:1684
payload	N/A
EventResponse	N/A
eventID	17d5c11c-1484-4f9a-8838-d4ab78b2332b
statusCode	200
reason	OK

F11 CallRedirectedEvent

Parameter Name	Parameter Value
requestId	0:1685
payload	N/A
Event	N/A
eventID	23adac70-b0a3-445f-bdb3-6f0d08408659
sequenceNumber	11
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	48d32994-4030-415b-ac19-53d93ff36245
channelId	77f0f20c-f278-48ff-b409-7ca31ed522dc
targetId	subscriberS1@broadworks.com

Parameter Name	Parameter Value
eventData	N/A
calls	N/A
call	N/A
callId	callhalf-5627:0
extTrackingId	14060905-2d7a-469e-a4e9-8c4d4ed823b0
personality	Terminator
state	Detached
remoteParty	N/A
name	subscriberS2Firstname subscriberS2Lastname
address	tel:1012
userId	subscriberS2@broadworks.com
userDN	tel:+15006001012;ext=1012
callType	Group
redirect	N/A
address	tel:1013
reason	transfer
redirectTime	1382035884452
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
startTime	1382035846972
answerTime	1382035848055
detachedTime	1382035884453
call	N/A
callId	callhalf-5627:1
extTrackingId	14060905-2d7a-469e-a4e9-8c4d4ed823b0
personality	Originator
state	Detached
remoteParty	N/A
name	subscribers3Firstname subscribers3Lastname
address	tel:1013
userId	subscribers3@broadworks.com
userDN	tel:+15006001013;ext=1013
callType	Group
redirect	N/A
address	tel:1012

Parameter Name	Parameter Value
reason	transfer
redirectTime	1382035884453
startTime	1382035884272
detachedTime	1382035884453

F12 EventResponse

Parameter Name	Parameter Value
requestId	0:1685
payload	N/A
EventResponse	N/A
eventID	23adac70-b0a3-445f-bdb3-6f0d08408659
statusCode	200
reason	OK

F13 HookStatusEvent

Parameter Name	Parameter Value
requestId	0:1686
payload	N/A
Event	N/A
eventID	3b8235dd-4a67-4125-be33-40f377b836c8
sequenceNumber	12
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	48d32994-4030-415b-ac19-53d93ff36245
channelId	77f0f20c-f278-48ff-b409-7ca31ed522dc
targetId	subscriberS1@broadworks.com
eventData	N/A
hookStatus	On-Hook

F14 EventResponse

Parameter Name	Parameter Value
requestId	0:1686
payload	N/A
EventResponse	N/A
eventID	3b8235dd-4a67-4125-be33-40f377b836c8
statusCode	200
reason	OK

9.1.11 Reconnect

This scenario illustrates a successful Reconnect. Call C1 is in the *Held* state while call C2 is in the *Remote Held* state when the remote application initiates a reconnect.

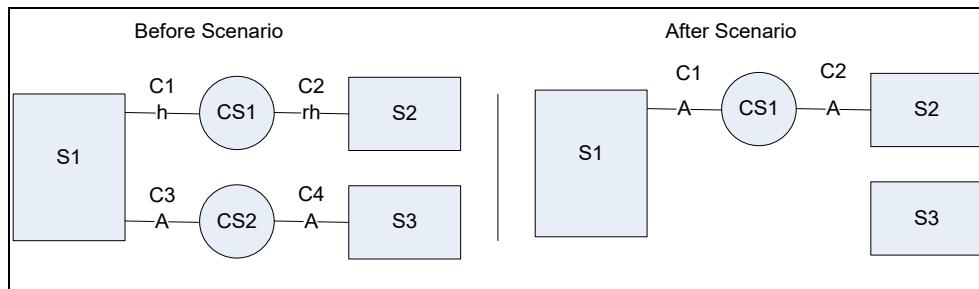


Figure 123 Reconnect Message Flow

Activity	Monitored Subscriber S1	Monitored Subscriber S2	Monitored Subscriber S3	Comments
A Reconnect Request is performed on behalf of subscriber S1.	Reconnect Request (F1) Reconnect Response (F2)			
Indication that calls C3 and C4 are released.	Call Released Event (F3) Event Response (F4)		Call Released Event (F5) Event Response (F6) Hook Status Event (F7) Event Response (F8)	
Indication that calls C1 and C2 are now back in the Active state.	Call Retrieved Event (F9) Event Response (F10)	Call Retrieved Event (F11) Event Response (F12)		

F1 Request

Parameter Name	Parameter Value
requestId	8
sessionId	16
credentials	YXNvdXRoQG10bGFzZGV2ODcubmV0Om10bGxhYg==
ReconnectRequest	N/A
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/%callId%/reconnect
method	PUT
version	N/A
params	N/A
callId	callhalf-5687:0
subscriberId	subscriberS1@broadworks.com



Parameter Name	Parameter Value
payload	N/A

F2 Response

Parameter Name	Parameter Value
requestId	8
sessionId	16
statusCode	200
reason	OK
payload	N/A

F3 CallReleasedEvent

Parameter Name	Parameter Value
requestId	0:1745
payload	N/A
Event	N/A
eventID	4200c390-0b33-4767-b606-73476021a6fe
sequenceNumber	13
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	dac75920-914e-4a68-998e-a99d182077d8
channelId	bb428283-ef18-421e-9fac-57d169ee96d1
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-5687:1
extTrackingId	1c241787-be79-471c-8fd1-054f09d4e686
personality	Originator
state	Released
releasingParty	localRelease
remoteParty	N/A
name	subscribers3Firstname subscribers3Lastname
address	tel:1013
userId	subscribers3@broadworks.com
userDN	tel:+15006001013;ext=1013
callType	Group
endpoint	N/A
addressOfRecord	5006001011@broadworks.com



Parameter Name	Parameter Value
startTime	1382036596855
answerTime	1382036598054
releaseTime	1382036639287

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:1745
payload	N/A
EventResponse	N/A
eventID	4200c390-0b33-4767-b606-73476021a6fe
statusCode	200
reason	OK

F5 CallReleasedEvent

Parameter Name	Parameter Value
requestId	0:1746
payload	N/A
Event	N/A
eventID	9d873f7e-af6a-49ba-8742-c45292ff112a
sequenceNumber	5
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	9da98157-87b3-4552-a699-b69c7c2a7990
channelId	bb428283-ef18-421e-9fac-57d169ee96d1
targetId	subscribers3@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-5695:0
extTrackingId	1c241787-be79-471c-8fd1-054f09d4e686
personality	Terminator
state	Released
releasingParty	remoteRelease
remoteParty	N/A
name	subscriberS1Firstname subscriberS1Lastname
address	tel:1011
userId	subscribers3@broadworks.com



Parameter Name	Parameter Value
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	N/A
addressOfRecord	5006001013@broadworks.com
startTime	1382036596855
answerTime	1382036598054
releaseTime	1382036639300

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:1746
payload	N/A
EventResponse	N/A
eventID	9d873f7e-af6a-49ba-8742-c45292ff112a
statusCode	200
reason	OK

F7 HookStatusEvent

Parameter Name	Parameter Value
requestId	0:1747
payload	N/A
Event	N/A
eventID	773edbb6-3b10-4ced-991e-a055f77fe094
sequenceNumber	6
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	9da98157-87b3-4552-a699-b69c7c2a7990
channelId	bb428283-ef18-421e-9fac-57d169ee96d1
targetId	subscribers3@broadworks.com
EventData	N/A
hookStatus	On-Hook

F8 EventResponse

Parameter Name	Parameter Value
requestId	0:1747
payload	N/A
EventResponse	N/A



Parameter Name	Parameter Value
eventID	773edbb6-3b10-4ced-991e-a055f77fe094
statusCode	200
reason	OK

F9 CallRetrievedEvent

Parameter Name	Parameter Value
requestId	0:1748
payload	N/A
Event	N/A
eventID	73c0b1a3-8149-43f6-8775-69a1d35d6275
sequenceNumber	14
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	dac75920-914e-4a68-998e-a99d182077d8
channelId	bb428283-ef18-421e-9fac-57d169ee96d1
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-5687:0
extTrackingId	815629c4-6171-4fee-8f7e-7863aa435e40
personality	Terminator
state	Active
remoteParty	N/A
name	subscriberS2Firstname subscriberS2Lastname
address	tel:1012
userId	subscriberS2@broadworks.com
userDN	tel:+15006001012;ext=1012
callType	Group
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
startTime	1382036523549
answerTime	1382036524757
totalHeldTime	73308



F10 EventResponse

Parameter Name	Parameter Value
requestId	0:1748
payload	N/A
EventResponse	N/A
eventID	73c0b1a3-8149-43f6-8775-69a1d35d6275
statusCode	200
reason	OK

F11 CallRetrievedEvent

Parameter Name	Parameter Value
requestId	0:1749
payload	N/A
Event	N/A
eventID	9e327f91-2ced-445f-9637-d1f3f5e41fc7
sequenceNumber	8
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	642ff9da-6280-46f9-980a-def02c2307f9
channelId	bb428283-ef18-421e-9fac-57d169ee96d1
targetId	subscriberS2@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-5683:0
extTrackingId	815629c4-6171-4fee-8f7e-7863aa435e40
personality	Originator
state	Active
remoteParty	N/A
name	subscriberS1Firstname subscriberS1Lastname
address	tel:1011
userId	subscriberS1@broadworks.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	N/A
addressOfRecord	5006001012@broadworks.com
appearance	1
startTime	1382036523549

Parameter Name	Parameter Value
answerTime	1382036524757

F12 EventResponse

Parameter Name	Parameter Value
requestId	0:1749
payload	N/A
EventResponse	N/A
eventID	9e327f91-2ced-445f-9637-d1f3f5e41fc7
statusCode	200
reason	OK

9.1.12 Transfer to Voice Mail

This scenario illustrates a successful Transfer to Voice Mail. Subscriber S1 and S2 calls are both in the *Active* state when the remote application initiates a Transfer to Voice Mail. The Voice Portal number is 1050. In this scenario, the call is transferred to subscriber S1 voice mail.

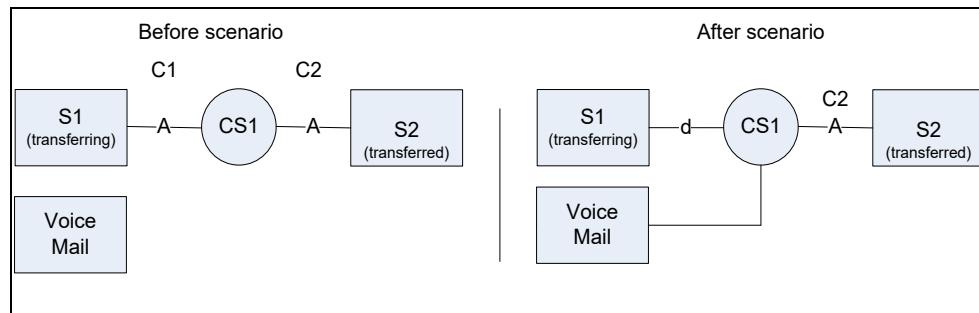


Figure 124 Transfer to Voice Mail Message Flow

Activity	Monitored Subscriber S1	Monitored Subscriber S2	Comments
A Transfer to Voice Mail Request to invoked on behalf of subscriber S1.	Transfer to Voice Mail Request (F1) Response (F2)		
Indication that the Request has been initiated from this subscriber.	Call Redirected Event (F3) Response (F4) Hook Status Event (F5) Event Response (F6)	Call Transferred Event (F7) Response (F8)	The Call Redirected event specifies that a redirection is performed to the Voice Mail.
The voice mail answers the call.		Call Answered Event (F9) Response (F10)	



F1 Request

Parameter Name	Parameter Value
requestId	27
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
TransferToOwnVoice Mail	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/%callId%/vmtransfer
method	PUT
version	Not applicable
params	Not applicable
callId	callhalf-63751:0
subscriberId	subscriberS1@broadsoft.com
payload	Not applicable

F2 Response

Parameter Name	Parameter Value
requestId	27
sessionId	245
statusCode	200
reason	OK
payload	Not applicable

F3 CallRedirectedEvent

Parameter Name	Parameter Value
requestId	21:511
payload	Not applicable
Event	Not applicable
eventID	c55df898-b945-4aa8-a5b0-70af3ea753f7
sequenceNumber	8
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	c9dae9c6-0e11-4993-bc9a-7498580e3c9b
channelId	04074f9c-b8f6-4431-9f80-4228a674a9b5
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
calls	Not applicable

Parameter Name	Parameter Value
call	Not applicable
callId	callhalf-63751:0
extTrackingId	527:1
personality	Terminator
state	Detached
remoteParty	Not applicable
name	subscriberS2FirstName subscriberS2LastName
address	tel:1012
userId	subscriberS2@broadsoft.com
userDN	tel:+15006001012;ext=1012
callType	Group
redirect	Not applicable
address	tel:1050
reason	transfer
redirectTime	1271797089050
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
startTime	1271797049472
answerTime	1271797050934
detachedTime	1271797089051

F4 EventResponse

Parameter Name	Parameter Value
requestId	21:511
payload	Not applicable
EventResponse	Not applicable
eventID	c55df898-b945-4aa8-a5b0-70af3ea753f7
statusCode	200
reason	OK

F5 HookStatusEvent

Parameter Name	Parameter Value
requestId	21:512
payload	N/A
Event	N/A
eventID	3b8235dd-4a67-4125-be33-40f377b836c8

Parameter Name	Parameter Value
sequenceNumber	9
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	c9dae9c6-0e11-4993-bc9a-7498580e3c9b
channelId	04074f9c-b8f6-4431-9f80-4228a674a9b5
targetId	subscriberS1@broadworks.com
eventData	N/A
hookStatus	On-Hook

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:1686
payload	N/A
EventResponse	N/A
eventID	3b8235dd-4a67-4125-be33-40f377b836c8
statusCode	200
reason	OK

F7 CallTransferredEvent

Parameter Name	Parameter Value
requestId	21:513
payload	Not applicable
Event	Not applicable
eventID	65b84566-91e5-4a44-b80f-2231484a8ab4
sequenceNumber	8
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	60c166b4-c965-489f-a989-9cebc478ba2
channelId	04074f9c-b8f6-4431-9f80-4228a674a9b5
targetId	subscriberS2@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-63747:0
extTrackingId	527:1
personality	Originator
state	Alerting

Parameter Name	Parameter Value
remoteParty	Not applicable
name	subscriberS1FirstName subscriberS1LastName
address	tel:1011
userId	subscriberS1@broadsoft.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	Not applicable
addressOfRecord	5006001012@broadsoft.com
appearance	1
startTime	1271797049472
answerTime	1271797050934

F8 EventResponse

Parameter Name	Parameter Value
requestId	21:513
payload	Not applicable
EventResponse	Not applicable
eventID	65b84566-91e5-4a44-b80f-2231484a8ab4
statusCode	200
reason	OK

F9 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	21:514
payload	Not applicable
Event	Not applicable
eventID	f06e073b-6ae6-4cc6-a352-9c476f9f14ff
sequenceNumber	9
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	60c166b4-c965-489f-a989-9cebc478ba2
channelId	04074f9c-b8f6-4431-9f80-4228a674a9b5
targetId	subscriberS2@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-63747:0

Parameter Name	Parameter Value
extTrackingId	527:1
personality	Originator
state	Active
remoteParty	Not applicable
name	subscriberS1FirstName subscriberS1LastName
address	tel:1011
userId	subscriberS1@broadsoft.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	Not applicable
addressOfRecord	5006001012@broadsoft.com
appearance	1
startTime	1271797049472
answerTime	1271797050934

F10 EventResponse

Parameter Name	Parameter Value
requestId	21:514
payload	Not applicable
EventResponse	Not applicable
eventID	f06e073b-6ae6-4cc6-a352-9c476f9f14ff
statusCode	200
reason	OK

9.1.13 Call Return (Two-level Mode)

This scenario illustrates a successful Call Return where the system is configured to work in a two-level mode that provides announcements to guide the user. Subscriber S1 and subscriber S2 are in the *Idle* state. Subscriber S2 was the last subscriber to call subscriber S1. Note that the last number redial scenario is very similar to this one without the confirmation part.

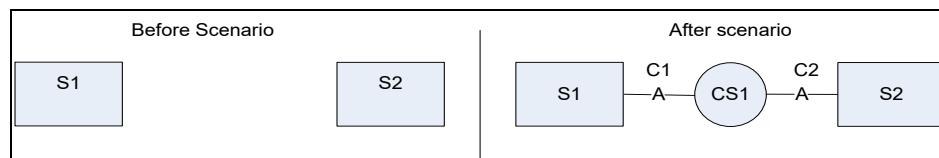


Figure 125 Call Return Message Flow



Activity	Monitored Subscriber S1	Monitored Subscriber S2	Comments
A Dial Request to a valid device is invoked on behalf of subscriber S1.	Call Return Request(F1) Response (F2)		The Dial Request specifies that subscriber S1 should be prompted to go off-hook. The response contains the newly created callId and externalTrackingId.
Indication that the service has been initiated from this subscriber.	Call Originating Event (F3) Response (F4)		The "Click-to-Dial" personality indicates that the subscriber S1 is being prompted (via ringing, for example) to go off-hook.
Indication that the subscriber's endpoint supports the SIP talk event package.	Call Updated Event (F5) Response (F6)		
Subscriber S1 goes off-hook. The last incoming number is announced and the subscriber is instructed to dial a configurable confirmation digit to activate Call Return. Subscriber S1 enters the confirmation DTMF through his keypad.	Hook Status Event (F7) Response (F8) Call Collecting Event (F9) Response (F10)		
The call attempt to S2 is initiated.	Call Originated Event (F11) Response (F12)		Call Personality is set to "Originator".
Subscriber S2 begins to ring and S1 receives ringing tone.		Call Received Event (F13) Response (F12)	
Subscriber S2 answers the call by manually going off-hook.	Call Answered Event (F17) Response (F18)	Hook Status Event (F15) Response (F16) Call Answered Event (F19) Response (F20)	Active for both the users.

F1 Request

Parameter Name	Parameter Value
requestId	6
sessionId	19
credentials	YXNvdXRoQG10bGFzZGV2ODcubmV0Om10bGxhYg==
CallReturnRequest	N/A
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/callreturn?%location%
method	POST
version	N/A



Parameter Name	Parameter Value
params	N/A
location	All
subscriberId	subscriberS1@broadworks.com
payload	N/A

F2 Response

Parameter Name	Parameter Value
requestId	6
sessionId	19
statusCode	201
reason	Created
payload	N/A
CallStartInfo	N/A
callId	callhalf-31:0
externalTrackingId	6b79d01e-d1a6-49bd-8de0-02510f7db6e0

F3 CallOriginatingEvent

Parameter Name	Parameter Value
requestId	0:1904
payload	N/A
Event	N/A
eventID	32317933-ee2e-40cd-89b2-c80b6937a3a2
sequenceNumber	15
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-31:0
extTrackingId	6b79d01e-d1a6-49bd-8de0-02510f7db6e0
personality	Click-to-Dial
state	Alerting
remoteParty	N/A
address	tel:*69



Parameter Name	Parameter Value
callType	Unknown
appearance	1
startTime	1382039509894

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:1904
payload	N/A
EventResponse	N/A
eventID	32317933-ee2e-40cd-89b2-c80b6937a3a2
statusCode	200
reason	OK

F5 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:1905
payload	N/A
Event	N/A
eventID	39c4f244-3895-4ecc-9af4-b12fa31686d1
sequenceNumber	16
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-31:0
extTrackingId	6b79d01e-d1a6-49bd-8de0-02510f7db6e0
personality	Click-to-Dial
state	Alerting
remoteParty	N/A
address	tel:*69
callType	Unknown
endpoint	N/A
addressOfRecord	5006001011@broadworks.com



Parameter Name	Parameter Value
appearance	1
allowAnswer	N/A
startTime	1382039509894

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:1905
payload	N/A
EventResponse	N/A
eventID	39c4f244-3895-4ecc-9af4-b12fa31686d1
statusCode	200
reason	OK

F7 HookStatusEvent

Parameter Name	Parameter Value
requestId	0:1906
payload	N/A
Event	N/A
eventID	64c4e5b3-e971-4d94-b9f0-53f8c3a2f67f
sequenceNumber	17
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
eventData	N/A
hookStatus	Off-Hook

F8 EventResponse

Parameter Name	Parameter Value
requestId	0:1906
payload	N/A
EventResponse	N/A
eventID	64c4e5b3-e971-4d94-b9f0-53f8c3a2f67f
statusCode	200
reason	OK



F9 CallCollectingEvent

Parameter Name	Parameter Value
requestId	0:1907
payload	N/A
Event	N/A
eventID	30ab3006-0e8a-4446-883b-a0e3a81b53db
sequenceNumber	18
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-31:0
extTrackingId	6b79d01e-d1a6-49bd-8de0-02510f7db6e0
personality	Originator
state	Alerting
remoteParty	N/A
address	tel:*69
callType	Unknown
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
startTime	1382039511612

F10 EventResponse

Parameter Name	Parameter Value
requestId	0:1907
payload	N/A
EventResponse	N/A
eventID	30ab3006-0e8a-4446-883b-a0e3a81b53db
statusCode	200
reason	OK



F11 CallOriginatedEvent

Parameter Name	Parameter Value
requestId	0:1909
payload	N/A
Event	N/A
eventID	9f884cd6-b224-48a2-bf0d-102d1f858596
sequenceNumber	19
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-31:0
extTrackingId	6b79d01e-d1a6-49bd-8de0-02510f7db6e0
personality	Originator
state	Alerting
remoteParty	N/A
name	subscriberS2Firstname subscriberS2Lastname
address	tel:1012
userId	subscriberS2@broadworks.com
userDN	tel:+15006001012;ext=1012
callType	Group
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
startTime	1382039511612

F12 EventResponse

Parameter Name	Parameter Value
requestId	0:1909
payload	N/A
EventResponse	N/A
eventID	9f884cd6-b224-48a2-bf0d-102d1f858596
statusCode	200
reason	OK

F13 CallReceivedEvent

Parameter Name	Parameter Value
requestId	0:1910
payload	N/A
Event	N/A
eventID	5d2e1ee7-5184-4e9b-9343-ecfc79e2b6db
sequenceNumber	9
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	e1b9b5c8-df67-4d9f-a73f-09c238017f49
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS2@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-35:0
extTrackingId	6b79d01e-d1a6-49bd-8de0-02510f7db6e0
personality	Terminator
state	Alerting
remoteParty	N/A
name	subscriberS1Firstname subscriberS1Lastname
address	tel:1011
userId	subscriberS1@broadworks.com
userDN	tel:+15006001011;ext=1011
callType	Group
startTime	1382039511612

F14 EventResponse

Parameter Name	Parameter Value
requestId	0:1910
payload	N/A
EventResponse	N/A
eventID	5d2e1ee7-5184-4e9b-9343-ecfc79e2b6db
statusCode	200
reason	OK

F15 HookStatusEvent

Parameter Name	Parameter Value
requestId	0:1911
payload	N/A
Event	N/A
eventID	a989dcfb-2c2a-46b3-9163-f7e88b189f64
sequenceNumber	10
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	e1b9b5c8-df67-4d9f-a73f-09c238017f49
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS2@broadworks.com
EventData	N/A
hookStatus	Off-Hook

F16 EventResponse

Parameter Name	Parameter Value
requestId	0:1911
payload	N/A
EventResponse	N/A
eventID	a989dcfb-2c2a-46b3-9163-f7e88b189f64
statusCode	200
reason	OK

F17 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:1912
payload	N/A
Event	N/A
eventID	05751eeb-dbfb-48dd-bba2-91550c9605da
sequenceNumber	20
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
EventData	N/A

Parameter Name	Parameter Value
call	N/A
callId	callhalf-31:0
extTrackingId	6b79d01e-d1a6-49bd-8de0-02510f7db6e0
personality	Originator
state	Active
remoteParty	N/A
name	subscriberS2Firstname subscriberS2Lastname
address	tel:1012
userId	subscriberS2@broadworks.com
userDN	tel:+15006001012;ext=1012
callType	Group
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
startTime	1382039511612
answerTime	1382039523686

F18 EventResponse

Parameter Name	Parameter Value
requestId	0:1912
payload	N/A
EventResponse	N/A
eventID	05751eeb-dbfb-48dd-bba2-91550c9605da
statusCode	200
reason	OK

F19 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:1913
payload	N/A
Event	N/A
eventID	c459a8f2-91c3-4bb0-a461-f37d7120b6fb
sequenceNumber	11
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	e1b9b5c8-df67-4d9f-a73f-09c238017f49

Parameter Name	Parameter Value
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS2@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-35:0
extTrackingId	6b79d01e-d1a6-49bd-8de0-02510f7db6e0
personality	Terminator
state	Active
remoteParty	N/A
name	subscriberS1Firstname subscriberS1Lastname
address	tel:1011
userId	subscriberS1@broadworks.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	N/A
addressOfRecord	5006001012@broadworks.com
appearance	1
startTime	1382039511612
answerTime	1382039523686

F20 EventResponse

Parameter Name	Parameter Value
requestId	0:1913
payload	N/A
EventResponse	N/A
eventId	c459a8f2-91c3-4bb0-a461-f37d7120b6fb
statusCode	200
reason	OK

9.1.14 Directed Call Pickup

This scenario illustrates a successful Directed Call Pickup. In this scenario, subscriber S3 initiates a call toward subscriber S2, which is picked up by subscriber S1.

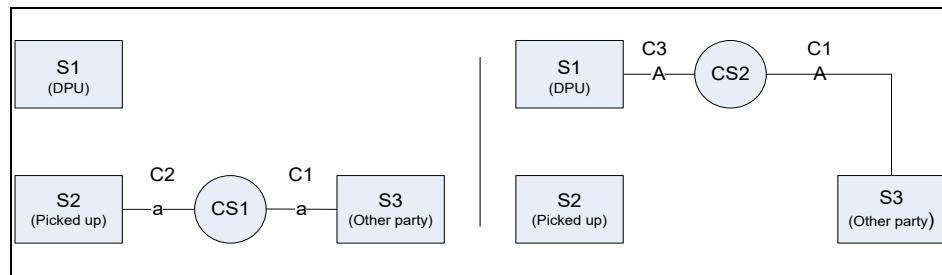


Figure 126 Directed Call Pickup Message Flow

Activity	Monitored Subscriber S1	Monitored Subscriber S2	Monitored Subscriber S3	Comments
A Direct Call Pickup is invoked on behalf of subscriber S1 specifying subscriber S2 phone number.	Directed Call Pickup Request (F1) Response (F2)			
Indication that the service has been initiated from this subscriber.	Call Originating Event (F3) Response (F4)			The "Click-to-Dial" personality indicates that the subscriber S1 is being prompted (via ringing, for example) to go off-hook.
Indication that the subscriber's endpoint supports the SIP talk event package.	Call Updated Event (F5) Event Response (F6)			
Subscriber S1 goes off-hook.	Hook Status Event (F7) Response (F8) Call Originated Event (F9) Response (F10)			
Indication that the call has been picked up.			Call Picked up Event (F11) Response (F12)	
Indication that the call to S2 is released as it has been picked up.		Call Released Event (F13) Response (F14)		
Call C3 personality is now Terminator.	Call Updated Event (F15) Response (F16)			
Indication that the call is answered.	Call Answered Event (F19) Response (F20)		Call Answered Event (F17) Response (F18)	



F1 Request

Parameter Name	Parameter Value
requestId	7
sessionId	19
credentials	YXNvdXRoQG10bGFzZGV2ODcubmV0Om10bGxhYg==
DirectedCallPickupRequest	N/A
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/directedcallpickup?%address%&%location%
method	POST
version	N/A
params	N/A
location	All
address	tel:1012
subscriberId	subscriberS1@broadworks.com
payload	N/A

F2 Response

Parameter Name	Parameter Value
requestId	7
sessionId	19
statusCode	201
reason	Created
payload	N/A
CallStartInfo	N/A
callId	callhalf-179:0
externalTrackingId	c9ce7a75-3b3b-4c93-892e-dc65e8a776e1

F3 CallOriginatingEvent

Parameter Name	Parameter Value
requestId	0:1985
payload	N/A
Event	N/A
eventID	c24278bc-f35c-4a85-82fb-aba9e9bf3116
sequenceNumber	23
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90

Parameter Name	Parameter Value
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-179:0
extTrackingId	c9ce7a75-3b3b-4c93-892e-dc65e8a776e1
personality	Click-to-Dial
state	Alerting
remoteParty	N/A
address	tel:*971012
callType	Unknown
appearance	1
startTime	1382041465142

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:1985
payload	N/A
EventResponse	N/A
eventId	c24278bc-f35c-4a85-82fb-aba9e9bf3116
statusCode	200
reason	OK

F5 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:1986
payload	N/A
Event	N/A
eventId	ae059fb9-1b4e-4685-ac0c-1f3c7a0adf76
sequenceNumber	24
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
eventData	N/A



Parameter Name	Parameter Value
call	N/A
callId	callhalf-179:0
extTrackingId	c9ce7a75-3b3b-4c93-892e-dc65e8a776e1
personality	Click-to-Dial
state	Alerting
remoteParty	N/A
address	tel:*971012
callType	Unknown
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
allowAnswer	N/A
startTime	1382041465142

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:1986
payload	N/A
EventResponse	N/A
eventID	ae059fb9-1b4e-4685-ac0c-1f3c7a0adf76
statusCode	200
reason	OK

F7 HookStatusEvent

Parameter Name	Parameter Value
requestId	0:1987
payload	N/A
Event	N/A
eventID	947b2422-b4f6-4abc-bcfa-e46555ec722f
sequenceNumber	25
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
EventData	N/A



Parameter Name	Parameter Value
hookStatus	Off-Hook

F8 EventResponse

Parameter Name	Parameter Value
requestId	0:1987
payload	N/A
EventResponse	N/A
eventID	947b2422-b4f6-4abc-bcfa-e46555ec722f
statusCode	200
reason	OK

F9 CallOriginatedEvent

Parameter Name	Parameter Value
requestId	0:1988
payload	N/A
Event	N/A
eventID	7cc8c43e-2807-4732-bb3b-4a4602df294b
sequenceNumber	26
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-179:0
extTrackingId	c9ce7a75-3b3b-4c93-892e-dc65e8a776e1
personality	Originator
state	Alerting
remoteParty	N/A
address	tel:*971012
callType	Unknown
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
startTime	1382041467282



F10 EventResponse

Parameter Name	Parameter Value
requestId	0:1988
payload	N/A
EventResponse	N/A
eventID	7cc8c43e-2807-4732-bb3b-4a4602df294b
statusCode	200
reason	OK

F11 CallPickedUpEvent

Parameter Name	Parameter Value
requestId	0:1990
payload	N/A
Event	N/A
eventID	3cf423b4-eab3-436a-9f22-7e5e430bd7a1
sequenceNumber	4
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	87dce460-e20c-4299-a341-5607792db688
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscribers3@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-171:0
extTrackingId	c9ce7a75-3b3b-4c93-892e-dc65e8a776e1
personality	Originator
state	Active
remoteParty	N/A
name	subscriberS1Firstname subscriberS1Lastname
address	tel:1011
userId	subscriberS1@broadworks.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	N/A
addressOfRecord	5006001013@broadworks.com
appearance	1
startTime	1382041454171



Parameter Name	Parameter Value
answerTime	1382041467434

F12 EventResponse

Parameter Name	Parameter Value
requestId	0:1990
payload	N/A
EventResponse	N/A
eventID	3cf423b4-eab3-436a-9f22-7e5e430bd7a1
statusCode	200
reason	OK

F13 CallReleasedEvent

Parameter Name	Parameter Value
requestId	0:1989
payload	N/A
Event	N/A
eventID	4a37f1ac-167c-430b-a068-957c3c04ed6d
sequenceNumber	15
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	e1b9b5c8-df67-4d9f-a73f-09c238017f49
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS2@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-175:0
extTrackingId	b5e842c8-03c6-4886-b63e-2e2d977f6f31
personality	Terminator
state	Released
releasingParty	remoteRelease
remoteParty	N/A
name	subscribers3Firstname subscribers3Lastname
address	tel:1013
userId	subscribers3@broadworks.com
userDN	tel:+15006001013;ext=1013
callType	Group

Parameter Name	Parameter Value
endpoint	N/A
addressOfRecord	5006001012@broadworks.com
startTime	1382041454171
releaseTime	1382041467448

F14 EventResponse

Parameter Name	Parameter Value
requestId	0:1989
payload	N/A
EventResponse	N/A
eventID	4a37f1ac-167c-430b-a068-957c3c04ed6d
statusCode	200
reason	OK

F15 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:1991
payload	N/A
Event	N/A
eventID	8908eec3-f72e-4b96-8892-d84dd6976e61
sequenceNumber	27
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-179:0
extTrackingId	c9ce7a75-3b3b-4c93-892e-dc65e8a776e1
personality	Terminator
state	Alerting
remoteParty	N/A
address	tel:*971012
callType	Unknown
endpoint	N/A



Parameter Name	Parameter Value
addressOfRecord	5006001011@broadworks.com
appearance	1
allowAnswer	N/A
startTime	1382041467282

F16 EventResponse

Parameter Name	Parameter Value
requestId	0:1991
payload	N/A
EventResponse	N/A
eventID	8908eec3-f72e-4b96-8892-d84dd6976e61
statusCode	200
reason	OK

F17 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:1992
payload	N/A
Event	N/A
eventID	557eff31-ef8f-4856-bd9b-1cdd4ff8cee3
sequenceNumber	5
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	87dce460-e20c-4299-a341-5607792db688
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscribers3@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-171:0
extTrackingId	c9ce7a75-3b3b-4c93-892e-dc65e8a776e1
personality	Originator
state	Active
remoteParty	N/A
name	subscriberS1Firstname subscriberS1Lastname
address	tel:1011
userId	subscriberS1@broadworks.com



Parameter Name	Parameter Value
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	N/A
addressOfRecord	5006001013@broadworks.com
appearance	1
startTime	1382041454171
answerTime	1382041467434

F18 EventResponse

Parameter Name	Parameter Value
requestId	0:1992
payload	N/A
EventResponse	N/A
eventID	557eff31-ef8f-4856-bd9b-1cdd4ff8cee3
statusCode	200
reason	OK

F19 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:1993
payload	N/A
Event	N/A
eventID	81ebf814-3627-471b-986e-618ed8567996
sequenceNumber	28
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-179:0
extTrackingId	c9ce7a75-3b3b-4c93-892e-dc65e8a776e1
personality	Terminator
state	Active
remoteParty	N/A

Parameter Name	Parameter Value
name	subscribers3Firstname subscribers3Lastname
address	tel:1013
userId	subscribers3@broadworks.com
userDN	tel:+15006001013;ext=1013
callType	Group
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
startTime	1382041467282
answerTime	1382041467434

F20 EventResponse

Parameter Name	Parameter Value
requestId	0:1993
payload	N/A
EventResponse	N/A
eventID	81ebf814-3627-471b-986e-618ed8567996
statusCode	200
reason	OK

9.1.15 Directed Call Pickup with Barge-in

This scenario illustrates a successful Directed Call Pickup with Barge-in (DPUBI). Subscriber S2 and S3 are both connected to each other in the *Active* state when the remote application initiates a DPUBI with subscriber S1.

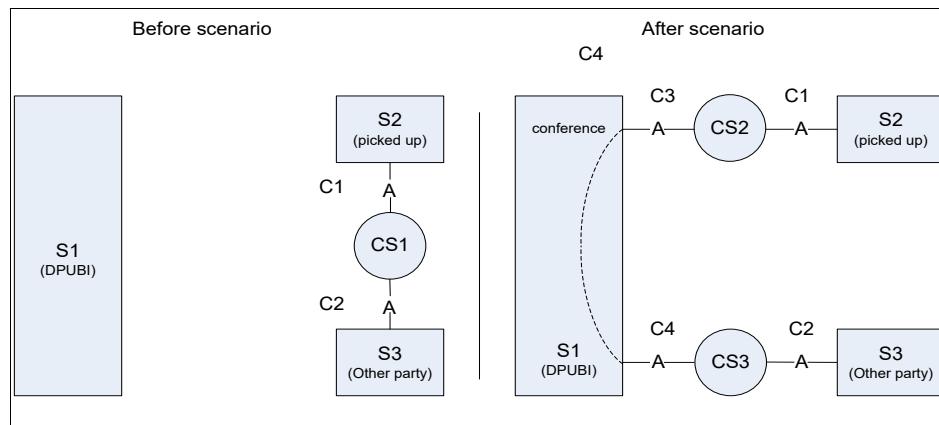


Figure 127 Directed Call Pickup with Barge-in Message Flow

Activity	Monitored Subscriber S1	Monitored Subscriber S2	Monitored Subscriber S3	Comments
A Direct Call Pickup with Barge-in is invoked on behalf of subscriber S1.	DPUBI Request (F1) Response (F2)			
Indication that the service has been initiated from this subscriber.	Call Originating Event (F3) Response (F4)			The "Click-to-Dial" personality indicates that the subscriber S1 is being prompted (via ringing, for example) to go off-hook.
Indication that the subscriber's endpoint supports the SIP talk event package.	Call Updated Event (F5) Response (F6)			
Subscriber S1 goes off-hook and the barge-in is initiated.	Hook Status Event (F7) Response (F8) Call Originated Event (F9) Response (F10)			C3 personality is Originator and is in the <i>Alerting</i> state.
Indication that call has been barged in.		Call Barged-in Event (F11) Response		This event specifies a new externalTrackingId (CS2).
Indication that call has been barged in.		Call Barged-in Event (F11) Response (F12)	Call Barged-in Event (F13) Response (F14)	This event specifies a new externalTrackingId (CS3).
Indicates that call C4 is originated.	Call Originated Event (F15) Response (F16)			
Indication that call C3 is now Active.	Call Answered Event (F17) Response (F18)			
Indicates that call C3 has no endpoint associated any longer (as a conference is to be started) and call C4 is the Originator.	Call Updated Event (F19) Response (F20) Call Updated Event (F21) Response (F22)			
Indicates that call C4 is now Active.	Call Answered Event (F23) Response (F24)			
Indicates that the conference is started.	Conference Started Event (F25) Response (F26)			

F1 Request

Parameter Name	Parameter Value
requestId	8
sessionId	19



Parameter Name	Parameter Value
credentials	YXNvdXRoQG10bGFzZGV2ODcubmV0Om10bGxhYg==
DirectedCallPickupWithBargeInRequest	N/A
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/directedcallpickupbargein?%address%&%location%
method	POST
version	N/A
params	N/A
location	All
address	tel:1012
subscriberId	subscriberS1@broadworks.com
payload	N/A

F2 Response

Parameter Name	Parameter Value
requestId	8
sessionId	19
statusCode	201
reason	Created
payload	N/A
CallStartInfo	N/A
callId	callhalf-327:0
externalTrackingId	cc15b170-3d34-4237-8566-8dd6663cfed3

F3 CallOriginatingEvent

Parameter Name	Parameter Value
requestId	0:2063
payload	N/A
Event	N/A
eventId	da482f15-95b7-4339-af1e-f65f09bb8c96
sequenceNumber	32
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
eventData	N/A



Parameter Name	Parameter Value
call	N/A
callId	callhalf-327:0
extTrackingId	cc15b170-3d34-4237-8566-8dd6663cfed3
personality	Click-to-Dial
state	Alerting
remoteParty	N/A
address	tel:*331012
callType	Unknown
appearance	1
startTime	1382043174087

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:2063
payload	N/A
EventResponse	N/A
eventID	da482f15-95b7-4339-af1e-f65f09bb8c96
statusCode	200
reason	OK

F5 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:2064
payload	N/A
Event	N/A
eventID	e402e523-b1f9-4e3e-9fdb-c4b0a920549f
sequenceNumber	33
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-327:0
extTrackingId	cc15b170-3d34-4237-8566-8dd6663cfed3



Parameter Name	Parameter Value
personality	Click-to-Dial
state	Alerting
remoteParty	N/A
address	tel:*331012
callType	Unknown
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
allowAnswer	N/A
startTime	1382043174087

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:2064
payload	N/A
EventResponse	N/A
eventID	e402e523-b1f9-4e3e-9fdb-c4b0a920549f
statusCode	200
reason	OK

F7 HookStatusEvent

Parameter Name	Parameter Value
requestId	0:2065
payload	N/A
Event	N/A
eventID	b3b086cd-1029-4eee-938b-da06b8ccd4dd
sequenceNumber	34
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
EventData	N/A
hookStatus	Off-Hook



F8 EventResponse

Parameter Name	Parameter Value
requestId	0:2065
payload	N/A
EventResponse	N/A
eventID	b3b086cd-1029-4eee-938b-da06b8ccd4dd
statusCode	200
reason	OK

F9 CallOriginatedEvent

Parameter Name	Parameter Value
requestId	0:2066
payload	N/A
Event	N/A
eventID	8f927002-1253-471e-a622-3604366c1c59
sequenceNumber	35
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-327:0
extTrackingId	cc15b170-3d34-4237-8566-8dd6663cfed3
personality	Originator
state	Alerting
remoteParty	N/A
address	tel:*331012
callType	Unknown
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
startTime	1382043175604



F10 EventResponse

Parameter Name	Parameter Value
requestId	0:2066
payload	N/A
EventResponse	N/A
eventID	8f927002-1253-471e-a622-3604366c1c59
statusCode	200
reason	OK

F11 CallBargedInEvent

Parameter Name	Parameter Value
requestId	0:2067
payload	N/A
Event	N/A
eventID	350ed5e3-dd33-4651-acfc-5da39899ee06
sequenceNumber	19
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	e1b9b5c8-df67-4d9f-a73f-09c238017f49
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS2@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-319:0
extTrackingId	cc15b170-3d34-4237-8566-8dd6663cfed3
personality	Terminator
state	Active
remoteParty	N/A
name	subscriberS1Firstname subscriberS1Lastname
address	tel:1011
userId	subscriberS1@broadworks.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	N/A
addressOfRecord	5006001012@broadworks.com
appearance	1
startTime	1382043175640



Parameter Name	Parameter Value
answerTime	1382043175640

F12 EventResponse

Parameter Name	Parameter Value
requestId	0:2067
payload	N/A
EventResponse	N/A
eventID	350ed5e3-dd33-4651-acfc-5da39899ee06
statusCode	200
reason	OK

F13 CallBargedInEvent

Parameter Name	Parameter Value
requestId	0:2070
payload	N/A
Event	N/A
eventID	7f0ceb51-a5a6-4e80-a809-7c4911080c0e
sequenceNumber	12
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	87dce460-e20c-4299-a341-5607792db688
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscribers3@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-315:0
extTrackingId	4b317d6e-a670-4597-8add-e0dde1916ce7
personality	Originator
state	Active
remoteParty	N/A
name	subscriberS1Firstname subscriberS1Lastname
address	tel:1011
userId	subscriberS1@broadworks.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	N/A



Parameter Name	Parameter Value
addressOfRecord	5006001013@broadworks.com
appearance	1
startTime	1382043175640
answerTime	1382043175640

F14 EventResponse

Parameter Name	Parameter Value
requestId	0:2070
payload	N/A
EventResponse	N/A
eventID	7f0ceb51-a5a6-4e80-a809-7c4911080c0e
statusCode	200
reason	OK

F15 CallOriginatedEvent

Parameter Name	Parameter Value
requestId	0:2068
payload	N/A
Event	N/A
eventID	9a7464af-565a-4766-81aa-15b924746cc8
sequenceNumber	36
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-327:1
extTrackingId	4b317d6e-a670-4597-8add-e0dde1916ce7
personality	Originator
state	Alerting
remoteParty	N/A
name	subscribers3Firstname subscribers3Lastname
address	tel:1013
userId	subscribers3@broadworks.com



Parameter Name	Parameter Value
userDN	tel:+15006001013;ext=1013
callType	Group
startTime	1382043175640

F16 EventResponse

Parameter Name	Parameter Value
requestId	0:2068
payload	N/A
EventResponse	N/A
eventID	9a7464af-565a-4766-81aa-15b924746cc8
statusCode	200
reason	OK

F17 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:2069
payload	N/A
Event	N/A
eventID	e7efcac7-c58e-4654-8f41-36aa06915dd2
sequenceNumber	37
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-327:0
extTrackingId	cc15b170-3d34-4237-8566-8dd6663cfed3
personality	Originator
state	Active
remoteParty	N/A
name	subscriberS2Firstname subscriberS2Lastname
address	tel:1012
userId	subscriberS2@broadworks.com
userDN	tel:+15006001012;ext=1012



Parameter Name	Parameter Value
callType	Group
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
startTime	1382043175604
answerTime	1382043175640

F18 EventResponse

Parameter Name	Parameter Value
requestId	0:2069
payload	N/A
EventResponse	N/A
eventID	e7efcac7-c58e-4654-8f41-36aa06915dd2
statusCode	200
reason	OK

F19 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:2071
payload	N/A
Event	N/A
eventID	27587404-843c-4d32-95f1-08dfe8d8a03d
sequenceNumber	38
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-327:1
extTrackingId	4b317d6e-a670-4597-8add-e0dde1916ce7
personality	Terminator
state	Alerting
remoteParty	N/A
name	subscribers3Firstname subscribers3Lastname



Parameter Name	Parameter Value
address	tel:1013
userId	subscribers3@broadworks.com
userDN	tel:+15006001013;ext=1013
callType	Group
startTime	1382043175640

F20 EventResponse

Parameter Name	Parameter Value
requestId	0:2071
payload	N/A
EventResponse	N/A
eventID	27587404-843c-4d32-95f1-08dfe8d8a03d
statusCode	200
reason	OK

F21 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:2072
payload	N/A
Event	N/A
eventID	34e998d6-d4d4-4c50-843f-6877a5c5bb3a
sequenceNumber	39
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-327:0
extTrackingId	cc15b170-3d34-4237-8566-8dd6663cfed3
personality	Originator
state	Active
remoteParty	N/A
name	subscriberS2Firstname subscriberS2Lastname
address	tel:1012



Parameter Name	Parameter Value
userId	subscriberS2@broadworks.com
userDN	tel:+15006001012;ext=1012
callType	Group
startTime	1382043175604
answerTime	1382043175640

F22 EventResponse

Parameter Name	Parameter Value
requestId	0:2072
payload	N/A
EventResponse	N/A
eventID	34e998d6-d4d4-4c50-843f-6877a5c5bb3a
statusCode	200
reason	OK

F23 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:2074
payload	N/A
Event	N/A
eventID	2bf68c79-3963-4f59-9657-97964de3e101
sequenceNumber	41
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-327:1
extTrackingId	4b317d6e-a670-4597-8add-e0dde1916ce7
personality	Terminator
state	Active
remoteParty	N/A
name	subscribers3Firstname subscribers3Lastname
address	tel:1013

Parameter Name	Parameter Value
userId	subscribers3@broadworks.com
userDN	tel:+15006001013;ext=1013
callType	Group
startTime	1382043175640
answerTime	1382043175640

F24 EventResponse

Parameter Name	Parameter Value
requestId	0:2074
payload	N/A
EventResponse	N/A
eventId	2bf68c79-3963-4f59-9657-97964de3e101
statusCode	200
reason	OK

F25 ConferenceStartedEvent

Parameter Name	Parameter Value
requestId	0:2073
payload	N/A
Event	N/A
eventId	5bf4b55b-5618-4a68-b364-74697e0716de
sequenceNumber	40
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d20acede-a246-46c2-9fac-ed3434eb6f90
channelId	f5d090fe-e0fc-4c9e-9fee-057fadd5580a
targetId	subscriberS1@broadworks.com
eventData	N/A
conference	N/A
state	Active
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
conferenceType	Barge-In
conferenceParticipantList	N/A
conferenceParticipant	N/A

Parameter Name	Parameter Value
callId	callhalf-327:0
conferenceParticipant	N/A
callId	callhalf-327:1
deaf	N/A

F26 EventResponse

Parameter Name	Parameter Value
requestId	0:2073
payload	N/A
EventResponse	N/A
eventID	5bf4b55b-5618-4a68-b364-74697e0716de
statusCode	200
reason	OK

9.1.16 Monitor Call (AS only)

This scenario illustrates a successful Monitor Call. In this scenario, agent A1 and subscriber S2 are connected through call session CS1 when subscriber S1 starts monitoring.

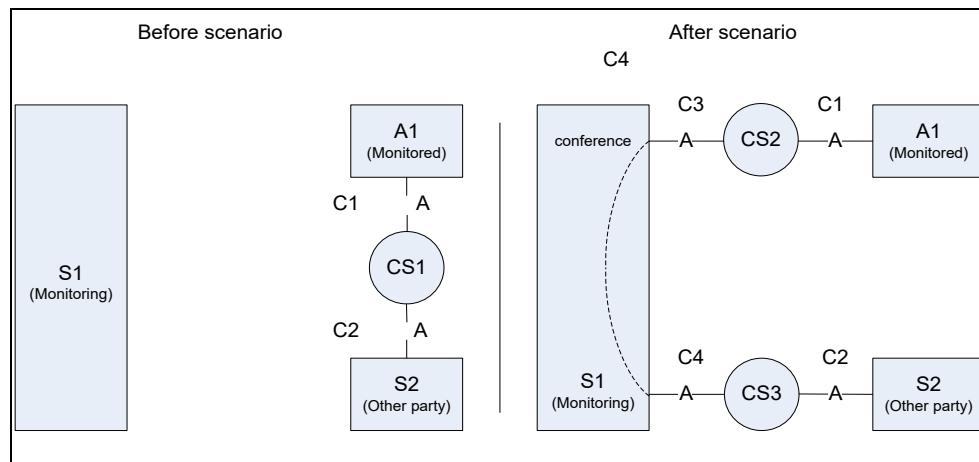


Figure 128 Monitor Call Message Flow

Activity	Monitored Subscriber S1	Monitored Agent A1	Monitored Subscriber S2	Comments
A Monitor Call is invoked on behalf of subscriber S1.	Monitor Call (F1) Response (F2)			
Indicates that the service has been initiated from this subscriber.	Call Originating Event (F3) Response (F4)			The “Click-to-Dial” personality indicates that the subscriber S1 is being prompted (via ringing, for example) to go off-hook.

Activity	Monitored Subscriber S1	Monitored Agent A1	Monitored Subscriber S2	Comments
	Call Updated Event (F5) Response (F6)			
Subscriber S1 goes off-hook and the monitoring is initiated.	Hook Status Event (F7) Response (F8) Call Originated Event (F9) Response (F10)			
Indicates that call C1 is being monitored.		Call Monitored Event (F11) Response (F12)		This event specifies a new externalTrackingId (CS2).
Indicates that call C2 is being monitored.			Call Monitored Event (F13) Response (F14)	This event specifies a new externalTrackingId (CS3).
Indicates that call C4 is created for the monitoring.	Call Originated Event (F15) Response (F16)			C4 personality is Originator and is in the <i>Held</i> state.
Indicates that call C3 is now Active.	Call Answered Event (F17) Response (F18)			
Indicates that call C3 has no endpoint associated any longer (as a conference is to be started).	Call Updated Event (F19) Response (F20)			
Indicates that call C4 is now the terminator.	Call Updated Event (F21) Response (F22)			
Indicates that the conference is started.	Conference Started Event (F23) Response (F24)			
Indicates that call C4 is now Active.	Call Answered Event (F25) Response (F26)			

F1 Request

Parameter Name	Parameter Value
requestId	15
sessionId	27
credentials	YXNvdXRoQG10bGFzZGV2ODcubmV0Om10bGxhYg==
MonitorCallRequest	N/A
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/silentmonitor?%address%&%location%



Parameter Name	Parameter Value
method	POST
version	N/A
params	N/A
location	All
address	tel:1020
subscriberId	subscriberS1@broadworks.com
payload	N/A

F2 Response

Parameter Name	Parameter Value
requestId	15
sessionId	27
statusCode	201
reason	Created
payload	N/A
CallStartInfo	N/A
callId	callhalf-30337:0
externalTrackingId	a238787d-b964-480c-8af0-750b24b01b6f

F3 CallOriginatingEvent

Parameter Name	Parameter Value
requestId	0:8150
payload	N/A
Event	N/A
eventID	e9eb6559-5434-4155-9a91-01ef2abf54c6
sequenceNumber	24
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	7a8eb65c-3e33-4260-a060-48c5129f000d
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-30337:0
extTrackingId	a238787d-b964-480c-8af0-750b24b01b6f
personality	Click-to-Dial



Parameter Name	Parameter Value
state	Alerting
remoteParty	N/A
address	tel:#821020
callType	Unknown
appearance	1
startTime	1382550646759

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:8150
payload	N/A
EventResponse	N/A
eventID	e9eb6559-5434-4155-9a91-01ef2abf54c6
statusCode	200
reason	OK

F5 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:8151
payload	N/A
Event	N/A
eventID	35fbe003-2b94-48b7-94a3-bacc3644b98c
sequenceNumber	25
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	7a8eb65c-3e33-4260-a060-48c5129f000d
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-30337:0
extTrackingId	a238787d-b964-480c-8af0-750b24b01b6f
personality	Click-to-Dial
state	Alerting
remoteParty	N/A
address	tel:#821020



Parameter Name	Parameter Value
callType	Unknown
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
allowAnswer	N/A
startTime	1382550646759

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:8151
payload	N/A
EventResponse	N/A
eventID	35fbe003-2b94-48b7-94a3-bacc3644b98c
statusCode	200
reason	OK

F7 HookStatusEvent

Parameter Name	Parameter Value
requestId	0:8152
payload	N/A
Event	N/A
eventID	8d7fa4df-18bd-4edb-8a1e-8cbc3a9139fa
sequenceNumber	26
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	7a8eb65c-3e33-4260-a060-48c5129f000d
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	subscriberS1@broadworks.com
eventData	N/A
hookStatus	Off-Hook

F8 EventResponse

Parameter Name	Parameter Value
requestId	0:8152
payload	N/A
EventResponse	N/A
eventID	8d7fa4df-18bd-4edb-8a1e-8cbc3a9139fa



Parameter Name	Parameter Value
statusCode	200
reason	OK

F9 CallOriginatedEvent

Parameter Name	Parameter Value
requestId	0:8154
payload	N/A
Event	N/A
eventID	8e50814c-59f2-4bd2-9c98-04f636ac4975
sequenceNumber	27
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	7a8eb65c-3e33-4260-a060-48c5129f000d
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-30337:0
extTrackingId	a238787d-b964-480c-8af0-750b24b01b6f
personality	Originator
state	Alerting
remoteParty	N/A
address	tel:#821020
callType	Unknown
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
startTime	1382550648222

F10 EventResponse

Parameter Name	Parameter Value
requestId	0:8154
payload	N/A
EventResponse	N/A
eventID	8e50814c-59f2-4bd2-9c98-04f636ac4975
statusCode	200



Parameter Name	Parameter Value
reason	OK

F11 CallMonitoredEvent

Parameter Name	Parameter Value
requestId	0:8153
payload	N/A
Event	N/A
eventID	b2691993-c03a-40f8-a254-c1ecc5ca2f38
sequenceNumber	71
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	eb134d07-4278-4a86-b499-b777c0e41c9b
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	agentA1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-30329:0
extTrackingId	a238787d-b964-480c-8af0-750b24b01b6f
personality	Terminator
state	Active
remoteParty	N/A
name	subscriberS2Firstname subscriberS2Lastname
address	tel:1012
userId	subscriberS2@broadworks.com
userDN	tel:+15006001012;ext=1012
callType	Group
endpoint	N/A
addressOfRecord	5006001020@broadworks.com
appearance	1
startTime	1382550648230
answerTime	1382550648230

F12 EventResponse

Parameter Name	Parameter Value
requestId	0:8153
payload	N/A



Parameter Name	Parameter Value
EventResponse	N/A
eventID	b2691993-c03a-40f8-a254-c1ecc5ca2f38
statusCode	200
reason	OK

F13 CallMonitoredEvent

Parameter Name	Parameter Value
requestId	0:8157
payload	N/A
Event	N/A
eventID	339f3a90-0032-4148-ad3c-abcb574319c0
sequenceNumber	5
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	d2824532-7293-44ff-9438-14e53a8be731
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	subscriberS2@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-30325:0
extTrackingId	c47ebf17-1c4d-4811-8580-286e8532b1d8
personality	Originator
state	Active
remoteParty	N/A
name	agentA1Firstname agentA1Lastname
address	tel:1020
userId	agentA1@broadworks.com
userDN	tel:+15006001020;ext=1020
callType	Group
endpoint	N/A
addressOfRecord	5006001012@broadworks.com
appearance	1
startTime	1382550648230
answerTime	1382550648230



F14 EventResponse

Parameter Name	Parameter Value
requestId	0:8157
payload	N/A
EventResponse	N/A
eventID	339f3a90-0032-4148-ad3c-abcb574319c0
statusCode	200
reason	OK

F15 CallOriginatedEvent

Parameter Name	Parameter Value
requestId	0:8155
payload	N/A
Event	N/A
eventID	c2c3158e-8002-4143-86fd-4900608b30d2
sequenceNumber	28
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	7a8eb65c-3e33-4260-a060-48c5129f000d
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-30337:1
extTrackingId	c47ebf17-1c4d-4811-8580-286e8532b1d8
personality	Originator
state	Alerting
remoteParty	N/A
name	subscriberS2Firstname subscriberS2Lastname
address	tel:1012
userId	subscriberS2@broadworks.com
userDN	tel:+15006001012;ext=1012
callType	Group
startTime	1382550648230



F16 EventResponse

Parameter Name	Parameter Value
requestId	0:8155
payload	N/A
EventResponse	N/A
eventID	c2c3158e-8002-4143-86fd-4900608b30d2
statusCode	200
reason	OK

F17 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:8156
payload	N/A
Event	N/A
eventID	de443914-5dd9-4514-ba62-2d1b80253f0d
sequenceNumber	29
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	7a8eb65c-3e33-4260-a060-48c5129f000d
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-30337:0
extTrackingId	a238787d-b964-480c-8af0-750b24b01b6f
personality	Originator
state	Active
remoteParty	N/A
name	agentA1Firstname agentA1Lastname
address	tel:1020
userId	agentA1@broadworks.com
userDN	tel:+15006001020;ext=1020
callType	Group
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
startTime	1382550648222



Parameter Name	Parameter Value
answerTime	1382550648230

F18 EventResponse

Parameter Name	Parameter Value
requestId	0:8156
payload	N/A
EventResponse	N/A
eventID	de443914-5dd9-4514-ba62-2d1b80253f0d
statusCode	200
reason	OK

F19 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:8158
payload	N/A
Event	N/A
eventID	aa1eace6-df43-41b9-9b8e-5c8da767b3f3
sequenceNumber	30
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	7a8eb65c-3e33-4260-a060-48c5129f000d
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-30337:1
extTrackingId	c47ebf17-1c4d-4811-8580-286e8532b1d8
personality	Terminator
state	Alerting
remoteParty	N/A
name	subscriberS2Firstname subscriberS2Lastname
address	tel:1012
userId	subscriberS2@broadworks.com
userDN	tel:+15006001012;ext=1012
callType	Group
startTime	1382550648230



F20 EventResponse

Parameter Name	Parameter Value
requestId	0:8158
payload	N/A
EventResponse	N/A
eventID	aa1eace6-df43-41b9-9b8e-5c8da767b3f3
statusCode	200
reason	OK

F21 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:8159
payload	N/A
Event	N/A
eventID	a3adecbd3-4302-4227-99e1-2a6dcf1c8b04
sequenceNumber	31
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	7a8eb65c-3e33-4260-a060-48c5129f000d
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-30337:0
extTrackingId	a238787d-b964-480c-8af0-750b24b01b6f
personality	Originator
state	Active
remoteParty	N/A
name	agentA1Firstname agentA1Lastname
address	tel:1020
userId	agentA1@broadworks.com
userDN	tel:+15006001020;ext=1020
callType	Group
startTime	1382550648222
answerTime	1382550648230



F22 EventResponse

Parameter Name	Parameter Value
requestId	0:8159
payload	N/A
EventResponse	N/A
eventID	a3adefbd3-4302-4227-99e1-2a6dcf1c8b04
statusCode	200
reason	OK

F23 ConferenceStartedEvent

Parameter Name	Parameter Value
requestId	0:8160
payload	N/A
Event	N/A
eventID	356db8ee-a7df-462b-b90d-333560664382
sequenceNumber	32
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	7a8eb65c-3e33-4260-a060-48c5129f000d
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	subscriberS1@broadworks.com
EventData	N/A
conference	N/A
state	Active
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
muted	N/A
conferenceType	Silently Monitored
conferenceParticipantList	N/A
conferenceParticipant	N/A
callId	callhalf-30337:0
conferenceParticipant	N/A
callId	callhalf-30337:1



F24 EventResponse

Parameter Name	Parameter Value
requestId	0:8160
payload	N/A
EventResponse	N/A
eventID	356db8ee-a7df-462b-b90d-333560664382
statusCode	200
reason	OK

F25 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:8161
payload	N/A
Event	N/A
eventID	95dbbcec-555a-48b2-b8ad-1daa8173d3e7
sequenceNumber	33
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	7a8eb65c-3e33-4260-a060-48c5129f000d
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-30337:1
extTrackingId	c47ebf17-1c4d-4811-8580-286e8532b1d8
personality	Terminator
state	Active
remoteParty	N/A
name	subscriberS2Firstname subscriberS2Lastname
address	tel:1012
userId	subscriberS2@broadworks.com
userDN	tel:+15006001012;ext=1012
callType	Group
startTime	1382550648230
answerTime	1382550648230

F26 EventResponse

Parameter Name	Parameter Value
requestId	0:8161
payload	N/A
EventResponse	N/A
eventId	95dbbcec-555a-48b2-b8ad-1daa8173d3e7
statusCode	200
reason	OK

9.2 Conference

Through the current section, the “conference” context has been abbreviated and replaced with “conf” for readability and presentation reasons.

The following subsections illustrate the following scenario:

- Conference start
- Conference call muted and made deaf
- Conference released
- Conference held
- Conference retrieved
- Conference add participant

9.2.1 Conference Start

This scenario illustrates a successful Conference Start. Subscriber S1 is the controller of the conference and has a call C1 in the *Held* state and a Call C3 in the Active state. In this scenario, a conference is established to join subscriber S2 and S3.

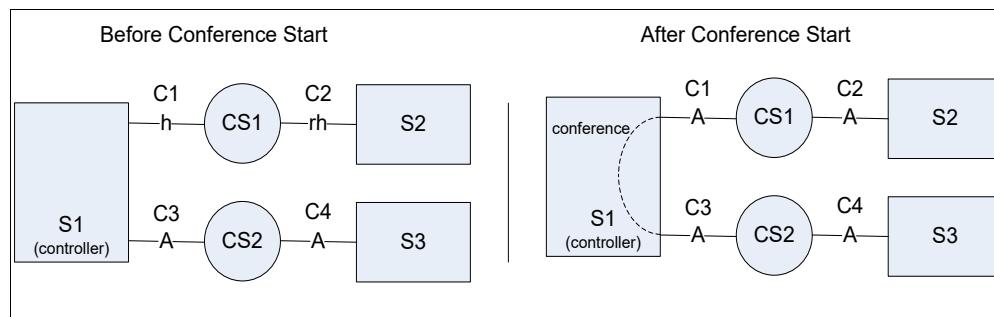


Figure 129 Conference Start Message Flow

Activity	Monitored Subscriber S1	Monitored Subscriber S2	Monitored Subscriber S3	Comments
Conference Call Service is requested on behalf of device S1.	Conference Start Request (F1) Response (F2)			



Activity	Monitored Subscriber S1	Monitored Subscriber S2	Monitored Subscriber S3	Comments
Indicates that the call C1 and C3 have no longer an endpoint associated with them.	Call Updated Event (F3 and F5) Response (F4 and F6)			
Indicates that the Request has been initiated from this subscriber.	Conference Started Event (F7) Response (F8)			
Indicates that Call C1 and C2 are now in Active state.	Call Retrieved Event (F9) Response (F10)	Call Retrieved Event (F11) Response (F12)		

F1 Request

Parameter Name	Parameter Value
requestId	9
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
ConferenceStart Request	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/conference
method	POST
version	Not applicable
params	Not applicable
subscriberId	subscriberS1@broadsoft.com
payload	Not applicable
Conference	Not applicable
conferenceParticipant List	Not applicable
conferenceParticipant	Not applicable
callId	callhalf-75923:0
conferenceParticipant	Not applicable
callId	callhalf-75923:1

F2 Response

Parameter Name	Parameter Value
requestId	9
sessionId	177
statusCode	200
reason	OK
payload	Not applicable



F3 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:1280
payload	Not applicable
Event	Not applicable
eventID	eba59e25-486b-4f75-9542-3333c7bd8b5c
sequenceNumber	42
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	66c6e8e4-090b-4be7-b2b7-32f507625eed
channelId	b62257b5-48b4-42c2-ad57-ffdc2182a3ca
targetId	subscriberS1@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-75923:0
extTrackingId	81:1
personality	Terminator
state	Held
remoteParty	Not applicable
name	subscriberS2FirstName subscriberS2LastName
address	tel:1012
userId	subscriberS2@broadsoft.com
userDN	tel:+15006001012;ext=1012
callType	Group
allowRetrieve	Not applicable
startTime	1271871781078
answerTime	1271871783697
heldTime	1271871788179

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:1280
payload	Not applicable
EventResponse	Not applicable
eventID	eba59e25-486b-4f75-9542-3333c7bd8b5c
statusCode	200
reason	OK

F5 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:1281
payload	Not applicable
Event	Not applicable
eventID	c9df68cd-6cc2-4dd0-ac59-852ba1986914
sequenceNumber	43
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	66c6e8e4-090b-4be7-b2b7-32f507625eed
channelId	b62257b5-48b4-42c2-ad57-ffdc2182a3ca
targetId	subscriberS1@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-75923:1
extTrackingId	82:1
personality	Terminator
state	Active
remoteParty	Not applicable
name	subscriberS3FirstName subscriberS3LastName
address	tel:1013
userId	subscriberS3@broadsoft.com
userDN	tel:+15006001013;ext=1013
callType	Group
startTime	1271871787090
answerTime	1271871788648

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:1281
payload	Not applicable
EventResponse	Not applicable
eventID	c9df68cd-6cc2-4dd0-ac59-852ba1986914
statusCode	200
reason	OK



F7 ConferenceStartedEvent

Parameter Name	Parameter Value
requestId	0:1283
payload	Not applicable
Event	Not applicable
eventide	542100ce-e2f5-4e21-8a76-53c86d97c9fd
sequenceNumber	44
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	66c6e8e4-090b-4be7-b2b7-32f507625eed
channelId	b62257b5-48b4-42c2-ad57-ffdc2182a3ca
targetId	subscriberS1@broadsoft.com
EventData	Not applicable
conference	Not applicable
state	Active
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
appearance	2
conferenceParticipantList	Not applicable
conferenceParticipant	Not applicable
callId	callhalf-75923:0
conferenceParticipant	Not applicable
callId	callhalf-75923:1

F8 EventResponse

Parameter Name	Parameter Value
requestId	0:1283
payload	Not applicable
EventResponse	Not applicable
eventide	542100ce-e2f5-4e21-8a76-53c86d97c9fd
statusCode	200
reason	OK

F9 CallRetrievedEvent

Parameter Name	Parameter Value
requestId	0:1284
payload	Not applicable



Parameter Name	Parameter Value
Event	Not applicable
eventid	636ab214-1ca1-4333-9997-3558ade44b15
sequenceNumber	45
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	66c6e8e4-090b-4be7-b2b7-32f507625eed
channelId	b62257b5-48b4-42c2-ad57-ffdc2182a3ca
targetId	subscriberS1@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-75923:0
extTrackingId	81:1
personality	Terminator
state	Active
remoteParty	Not applicable
name	subscriberS2FirstName subscriberS2LastName
address	tel:1012
userId	subscriberS2@broadsoft.com
userDN	tel:+15006001012;ext=1012
callType	Group
startTime	1271871781078
answerTime	1271871783697
totalHeldTime	53603

F10 EventResponse

Parameter Name	Parameter Value
requestId	0:1284
payload	Not applicable
EventResponse	Not applicable
eventID	636ab214-1ca1-4333-9997-3558ade44b15
statusCode	200
reason	OK

F11 CallRetrievedEvent

Parameter Name	Parameter Value
requestId	0:1282



Parameter Name	Parameter Value
payload	Not applicable
Event	Not applicable
eventID	49eee977-752b-4e0c-aaf9-cffdd39563b4
sequenceNumber	12
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	dfa0649f-331a-4a95-a0fe-4f4f528f0b9d
channelId	b62257b5-48b4-42c2-ad57-ffdc2182a3ca
targetId	subscriberS2@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-75919:0
extTrackingId	81:1
personality	Originator
state	Active
remoteParty	Not applicable
name	subscriberS1FirstName subscriberS1LastName
address	tel:1011
userId	subscriberS1@broadsoft.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	Not applicable
addressOfRecord	5006001012@broadsoft.com
appearance	1
startTime	1271871781078
answerTime	1271871783697

F12 EventResponse

Parameter Name	Parameter Value
requestId	0:1282
payload	Not applicable
EventResponse	Not applicable
eventID	49eee977-752b-4e0c-aaf9-cffdd39563b4
statusCode	200
reason	OK

9.2.2 Conference Call Muted and Made Deaf

This scenario illustrates a successful Conference Mute, Unmute, Deaf and Undeaf call. Subscriber S1 is the controller of the conference and has a call C1 and C3 in the *Active* state participating in a conference.

This scenario does not involve any state or topology changes, as illustrated in the following diagram.

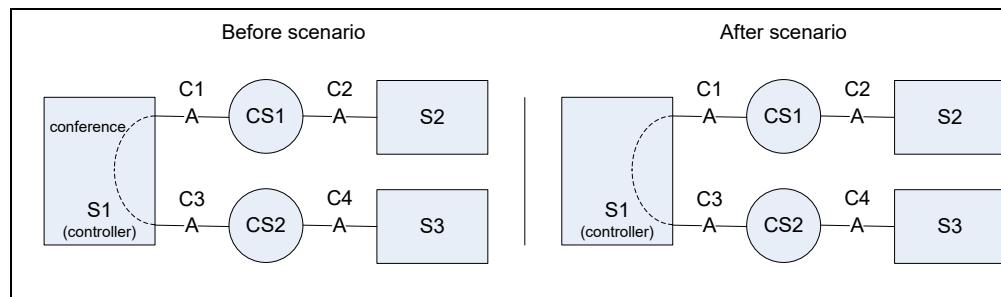


Figure 130 Conference Call Muted and Made Deaf Message Flow

Activity	Monitored Subscriber S1	Comments
Conference Mute Call is requested on behalf of device S1 for call C1.	Mute Call Request (F1) Mute Call Response (F2)	
Indication that the Call C1 is now muted.	Conference Call Muted Event (F3) Event Response (F4)	The audio stream from subscriber S2 is no longer added to the conference. S2 can hear the conference audio stream however.
Conference Unmute Call is requested on behalf of device S1 for call C1.	Unmute Call Request (F5) Unmute Call Response (F6)	
Indication that the Call C1 is no longer muted.	Conference Call Unmuted Event (F7) Event Response (F8)	The audio stream from and to subscriber S2 is back to normal.
Conference Deaf Call is requested on behalf of device S1 for call C1.	Deaf Call Request (F9) Deaf Call Response (F10)	
Indication that the Call C1 is now deaf.	Conference Call Made Deaf Event (F11) Event Response (F12)	The audio stream from the conference is no longer sent to subscriber S2. S2 can talk without hearing the conference audio stream however.
Conference Undeaf Call is requested on behalf of device S1 for call C1.	Undeaf Call Request (F13) Undeaf Call Response (F14)	
Indication that the Call C1 is no longer deaf.	Conference Call Made Undeaf Event (F15) Event Response (F16)	The audio stream from and to subscriber S2 is back to normal.

F1 Request

Parameter Name	Parameter Value
requestId	14
sessionId	MySessionId



Parameter Name	Parameter Value
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
ConferenceMuteRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/conference/%callId%/mute
method	PUT
version	Not applicable
params	Not applicable
callId	callhalf-23865:0
subscriberId	subscriberS1@broadsoft.com
payload	Not applicable

F2 Response

Parameter Name	Parameter Value
requestId	14
sessionId	50
statusCode	200
reason	OK
payload	Not applicable

F3 ConferenceCallMutedEvent

Parameter Name	Parameter Value
requestId	9:107
payload	Not applicable
Event	Not applicable
eventId	87d93632-12ab-42fd-b0ed-a569ea3d85e8
sequenceNumber	3
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	5c43fc11-ca31-4142-9756-ff8559c4eb9d
channelId	d7de7b9b-6335-441a-aba6-b2d7dbd4dc74
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
conference	Not applicable
state	Active
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com

Parameter Name	Parameter Value
appearance	2
conferenceParticipantList	Not applicable
conferenceParticipant	Not applicable
callId	callhalf-23865:0
muted	Not applicable
conferenceParticipant	Not applicable
callId	callhalf-23865:1

F4 EventResponse

Parameter Name	Parameter Value
requestId	9:107
payload	Not applicable
EventResponse	Not applicable
eventID	87d93632-12ab-42fd-b0ed-a569ea3d85e8
statusCode	200
reason	OK

F5 Request

Parameter Name	Parameter Value
requestId	15
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
ConferenceUnMuteRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/conference/%callId%/unmute
method	PUT
version	Not applicable
params	Not applicable
callId	callhalf-23865:0
subscriberId	subscriberS1@broadsoft.com
payload	Not applicable

F6 Response

Parameter Name	Parameter Value
requestId	15
sessionId	51



Parameter Name	Parameter Value
statusCode	200
reason	OK
payload	Not applicable

F7 ConferenceCallUnMutedEvent

Parameter Name	Parameter Value
requestId	9:108
payload	Not applicable
Event	Not applicable
eventID	80ee175c-be9e-48bc-be51-d89a6c339a48
sequenceNumber	4
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	5c43fc11-ca31-4142-9756-ff8559c4eb9d
channelId	d7de7b9b-6335-441a-aba6-b2d7dbd4dc74
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
conference	Not applicable
state	Active
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
appearance	2
conferenceParticipantList	Not applicable
conferenceParticipant	Not applicable
callId	callhalf-23865:0
conferenceParticipant	Not applicable
callId	callhalf-23865:1

F8 EventResponse

Parameter Name	Parameter Value
requestId	9:108
payload	Not applicable
EventResponse	Not applicable
eventID	80ee175c-be9e-48bc-be51-d89a6c339a48
statusCode	200
reason	OK



F9 Request

Parameter Name	Parameter Value
requestId	16
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
ConferenceDeaf Request	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/conference/%callId%/deaf
method	PUT
version	Not applicable
params	Not applicable
callId	callhalf-23865:0
subscriberId	subscriberS1@broadsoft.com
payload	Not applicable

F10 Response

Parameter Name	Parameter Value
requestId	16
sessionId	52
statusCode	200
reason	OK
payload	Not applicable

F11 ConferenceCallMadeDeafEvent

Parameter Name	Parameter Value
requestId	9:109
payload	Not applicable
Event	Not applicable
eventID	691af518-0ab3-462d-a724-6dabadfa70be
sequenceNumber	5
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	5c43fc11-ca31-4142-9756-ff8559c4eb9d
channelId	d7de7b9b-6335-441a-aba6-b2d7dbd4dc74
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
conference	Not applicable



Parameter Name	Parameter Value
state	Active
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
appearance	2
conferenceParticipantList	Not applicable
conferenceParticipant	Not applicable
callId	callhalf-23865:0
deaf	Not applicable
conferenceParticipant	Not applicable
callId	callhalf-23865:1

F12 EventResponse

Parameter Name	Parameter Value
requestId	9:109
payload	Not applicable
EventResponse	Not applicable
eventID	691af518-0ab3-462d-a724-6dabadfa70be
statusCode	200
reason	OK

F13 Request

Parameter Name	Parameter Value
requestId	17
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
ConferenceUnDeafRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/conference/%callId%/undeaf
method	PUT
version	Not applicable
params	Not applicable
callId	callhalf-23865:0
subscriberId	subscriberS1@broadsoft.com
payload	Not applicable



F14 Response

Parameter Name	Parameter Value
requestId	17
sessionId	53
statusCode	200
reason	OK
payload	Not applicable

F15 ConferenceCallMadeUnDeafEvent

Parameter Name	Parameter Value
requestId	9:110
payload	Not applicable
Event	Not applicable
eventID	df87802e-119b-439f-b220-d2178c7eb74b
sequenceNumber	6
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	5c43fc11-ca31-4142-9756-ff8559c4eb9d
channelId	d7de7b9b-6335-441a-aba6-b2d7dbd4dc74
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
conference	Not applicable
state	Active
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
appearance	2
conferenceParticipantList	Not applicable
conferenceParticipant	Not applicable
callId	callhalf-23865:0
conferenceParticipant	Not applicable
callId	callhalf-23865:1

F16 EventResponse

Parameter Name	Parameter Value
requestId	9:110
payload	Not applicable
EventResponse	Not applicable

Parameter Name	Parameter Value
eventID	df87802e-119b-439f-b220-d2178c7eb74b
statusCode	200
reason	OK

9.2.3 Conference Released

This scenario illustrates a successful Conference Release. Subscriber S1 is the controller of the conference and has calls C1 and C3 in the *Active* state participating in a conference. In this scenario, a conference is released and all calls relating to S1 are terminated.

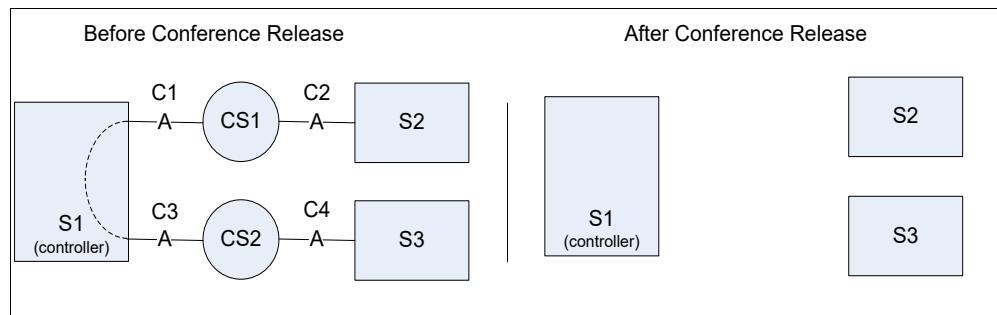


Figure 131 Conference Released Message Flow

Activity	Monitored Subscriber S1	Monitored Subscriber S2	Monitored Subscriber S3	Comments
Conference Release is requested on behalf of device S1.	Conference Release Request (F1) Response (F2)			
Indication that the Request has been initiated from this subscriber.	Conference Released Event (F3) Response(F4)			
Indication that the various calls have also been released.	Call Updated (F9) Response (F10) Call Released (F15) Response (F16) Call Released (F17) Response (F18) Hook Status Event (F19) Response (F20)	Call Released (F7) Response (F8) Hook Status Event (F11) Response (F12)	Call Released (F5) Response (F6) Hook Status Event (F11) Response (F12)	

F1 Request

Parameter Name	Parameter Value
requestId	5
sessionId	22
credentials	YXNvdXRoQG10bGFzZGV2ODcubmV0Om10bGxhYg==



Parameter Name	Parameter Value
ConferenceReleaseRequest	N/A
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%calls/conference
method	DELETE
version	N/A
params	N/A
subscriberId	subscriberS1@mtlasdev87.net
payload	N/A

F2 Response

Parameter Name	Parameter Value
requestId	5
sessionId	22
statusCode	200
reason	OK
payload	N/A

F3 ConferenceReleasedEvent

Parameter Name	Parameter Value
requestId	0:5001
payload	N/A
Event	N/A
eventID	1ddb2344-1baa-4f93-99c5-ffafc243d73c
sequenceNumber	14
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	fc0060f2-a74d-4f51-978c-82bd14626412
channelId	67e19868-9cef-4abe-ba5d-ffe00865589b
targetId	subscriberS1@broadworks.com
EventData	N/A
conference	N/A
state	Released

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:5001
payload	N/A



Parameter Name	Parameter Value
EventResponse	N/A
eventID	1ddb2344-1baa-4f93-99c5-ffafc243d73c
statusCode	200
reason	OK

F5 CallReleasedEvent

Parameter Name	Parameter Value
requestId	0:5002
payload	N/A
Event	N/A
eventID	91404bbf-5be8-47da-8fbf-d8c58ca1186b
sequenceNumber	5
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	f8645e9f-667d-456a-ba14-125b0464a77f
channelId	67e19868-9cef-4abe-ba5d-fde00865589b
targetId	subscriberS3@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-16779:0
extTrackingId	648b5421-0b51-4744-96c3-11b5d3f83297
personality	Originator
state	Released
releasingParty	remoteRelease
remoteParty	N/A
name	subscriberS1Firstname subscriberS1Lastname
address	tel:1011
userId	subscriberS1@broadworks.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	N/A
addressOfRecord	5006001013@broadworks.com
startTime	1382365726526
answerTime	1382365731971
releaseTime	1382365759046



F6 EventResponse

Parameter Name	Parameter Value
requestId	0:5002
payload	N/A
EventResponse	N/A
eventID	91404bbf-5be8-47da-8fbf-d8c58ca1186b
statusCode	200
reason	OK

F7 CallReleasedEvent

Parameter Name	Parameter Value
requestId	0:5003
payload	N/A
Event	N/A
eventID	15a40910-c464-4a78-b48d-17bf5fab44c4
sequenceNumber	7
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	f9ec56de-34f1-4e36-8b6e-da6b5fa0e4f8
channelId	67e19868-9cef-4abe-ba5d-ffe00865589b
targetId	subscriberS2@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-16771:0
extTrackingId	49d6aca9-aa11-48b6-bbd0-80989286ad29
personality	Originator
state	Released
releasingParty	remoteRelease
remoteParty	N/A
name	subscriberS1Firstname subscriberS1Lastname
address	tel:1011
userId	subscriberS1@broadworks.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	N/A
addressOfRecord	5006001012@broadworks.com
startTime	1382365721095



Parameter Name	Parameter Value
answerTime	1382365722541
releaseTime	1382365759048

F8 EventResponse

Parameter Name	Parameter Value
requestId	0:5003
payload	N/A
EventResponse	N/A
eventID	15a40910-c464-4a78-b48d-17bf5fab44c4
statusCode	200
reason	OK

F9 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:5004
payload	N/A
Event	N/A
eventID	3a75c71f-5c54-44aa-857f-b4aa8886263e
sequenceNumber	15
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	fc0060f2-a74d-4f51-978c-82bd14626412
channelId	67e19868-9cef-4abe-ba5d-fde00865589b
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-16775:1
extTrackingId	648b5421-0b51-4744-96c3-11b5d3f83297
personality	Terminator
state	Active
remoteParty	N/A
name	subscriberS3Firstname subscriberS3Lastname
address	tel:1013
userId	subscriberS3@broadworks.com
userDN	tel:+15006001013;ext=1013
callType	Group



Parameter Name	Parameter Value
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	2
startTime	1382365726526
answerTime	1382365731971

F10 EventResponse

Parameter Name	Parameter Value
requestId	0:5004
payload	N/A
EventResponse	N/A
eventID	3a75c71f-5c54-44aa-857f-b4aa8886263e
statusCode	200
reason	OK

F11 HookStatusEvent

Parameter Name	Parameter Value
requestId	0:5005
payload	N/A
Event	N/A
eventID	9910bbd5-53c1-4be0-a482-103352c6a355
sequenceNumber	6
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	f8645e9f-667d-456a-ba14-125b0464a77f
channelId	67e19868-9cef-4abe-ba5d-fde00865589b
targetId	subscriberS3@broadworks.com
EventData	N/A
hookStatus	On-Hook

F12 EventResponse

Parameter Name	Parameter Value
requestId	0:5005
payload	N/A
EventResponse	N/A
eventID	9910bbd5-53c1-4be0-a482-103352c6a355
statusCode	200



Parameter Name	Parameter Value
reason	OK

F13 HookStatusEvent

Parameter Name	Parameter Value
requestId	0:5006
payload	N/A
Event	N/A
eventID	c34d04d1-e675-4520-af5b-6018eb672a89
sequenceNumber	8
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	f9ec56de-34f1-4e36-8b6e-da6b5fa0e4f8
channelId	67e19868-9cef-4abe-ba5d-ffe00865589b
targetId	subscriberS2@broadworks.com
eventData	N/A
hookStatus	On-Hook

F14 EventResponse

Parameter Name	Parameter Value
requestId	0:5006
payload	N/A
EventResponse	N/A
eventID	c34d04d1-e675-4520-af5b-6018eb672a89
statusCode	200
reason	OK

F15 CallReleasedEvent

Parameter Name	Parameter Value
requestId	0:5007
payload	N/A
Event	N/A
eventID	c581fa6e-3c40-474c-bae3-d8d2ae8314ad
sequenceNumber	16
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	fc0060f2-a74d-4f51-978c-82bd14626412
channelId	67e19868-9cef-4abe-ba5d-ffe00865589b

Parameter Name	Parameter Value
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-16775:1
extTrackingId	648b5421-0b51-4744-96c3-11b5d3f83297
personality	Terminator
state	Released
releasingParty	localRelease
remoteParty	N/A
name	subscriberS3Firstname subscriberS3Lastname
address	tel:1013
userId	subscriberS3@broadworks.com
userDN	tel:+15006001013;ext=1013
callType	Group
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
startTime	1382365726526
answerTime	1382365731971
releaseTime	1382365759046

F16 EventResponse

Parameter Name	Parameter Value
requestId	0:5007
payload	N/A
EventResponse	N/A
eventID	c581fa6e-3c40-474c-bae3-d8d2ae8314ad
statusCode	200
reason	OK

F17 CallReleasedEvent

Parameter Name	Parameter Value
requestId	0:5008
payload	N/A
Event	N/A
eventID	689a1bcb-f5b1-4b7e-a1d9-28f1d0b87a6c
sequenceNumber	17



Parameter Name	Parameter Value
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	fc0060f2-a74d-4f51-978c-82bd14626412
channelId	67e19868-9cef-4abe-ba5d-ffe00865589b
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-16775:0
extTrackingId	49d6aca9-aa11-48b6-bbd0-80989286ad29
personality	Terminator
state	Released
releasingParty	localRelease
remoteParty	N/A
name	subscriberS2Firstname subscriberS2Lastname
address	tel:1012
userId	subscriberS2@broadworks.com
userDN	tel:+15006001012;ext=1012
callType	Group
startTime	1382365721095
answerTime	1382365722541
totalHeldTime	19353
releaseTime	1382365759046

F18 EventResponse

Parameter Name	Parameter Value
requestId	0:5008
payload	N/A
EventResponse	N/A
eventId	689a1bcb-f5b1-4b7e-a1d9-28f1d0b87a6c
statusCode	200
reason	OK

F19 HookStatusEvent

Parameter Name	Parameter Value
requestId	0:5009
payload	N/A



Parameter Name	Parameter Value
Event	N/A
eventID	395672c9-6055-4c04-92a8-ef8dbca00912
sequenceNumber	18
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	fc0060f2-a74d-4f51-978c-82bd14626412
channelId	67e19868-9cef-4abe-ba5d-ffe00865589b
targetId	subscriberS1@broadworks.com
eventData	N/A
hookStatus	On-Hook

F20 EventResponse

Parameter Name	Parameter Value
requestId	0:5009
payload	N/A
EventResponse	N/A
eventID	395672c9-6055-4c04-92a8-ef8dbca00912
statusCode	200
reason	OK

9.2.4 Conference Held

This scenario illustrates a success conference hold. Notice that the conference is in the *Held* state while all the calls are still in the *Active* state. S2 and S3 can talk to each other while the conference is held.

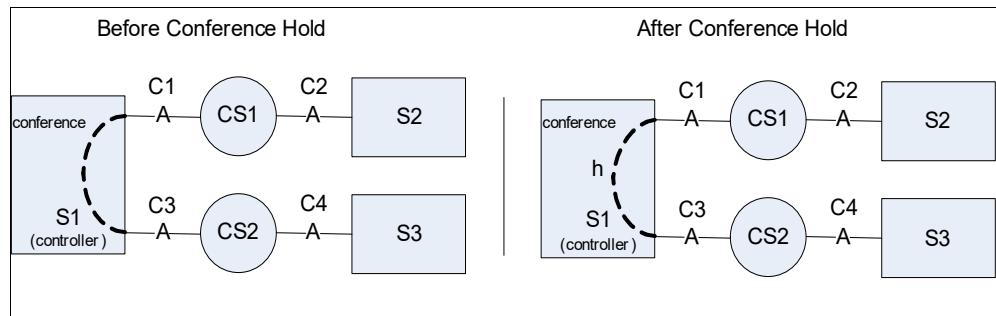


Figure 132 Conference Held Message Flow

Activity	Monitored Subscriber S1	Monitored Subscriber S2	Monitored Subscriber S3	Comments
Conference Hold is requested on behalf of device S1.	Conference Hold Request (F1) Conference Hold Response (F2)			
Indication that the conference is now held for subscriber S1.	Conference Held Event (F3) Event Response (F4)			S3 and S2 can still talk to each other.

F1 Request

Parameter Name	Parameter Value
requestId	23
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
ConferenceHold Request	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/conference/hold
method	PUT
version	Not applicable
params	Not applicable
subscriberId	subscriberS1@broadsoft.com
payload	Not applicable



F2 Response

Parameter Name	Parameter Value
requestId	23
sessionId	73
statusCode	200
reason	OK
payload	Not applicable

F3 ConferenceHeldEvent

Parameter Name	Parameter Value
requestId	9:192
payload	Not applicable
Event	Not applicable
eventID	e4608d7c-1154-4e52-afc4-dc07ac382e81
sequenceNumber	13
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	baae6a39-cf9f-4420-91a4-8dd734248176
channelId	d7de7b9b-6335-441a-aba6-b2d7dbd4dc74
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
conference	Not applicable
state	Held
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
appearance	2
Conference ParticipantList	Not applicable
Conference Participant	Not applicable
callId	callhalf-24131:0
Conference Participant	Not applicable
callId	callhalf-24131:1

F4 EventResponse

Parameter Name	Parameter Value
requestId	9:192
payload	Not applicable

Parameter Name	Parameter Value
EventResponse	Not applicable
eventID	e4608d7c-1154-4e52-afc4-dc07ac382e81
statusCode	200
reason	OK

9.2.5 Conference Retrieved

This scenario illustrates a successful Conference Retrieve. Subscriber S1 is the controller of the conference and has calls C1 and C3 in the *Active* state participating in a conference. In this scenario, a conference is released and all calls related to S1 are terminated.

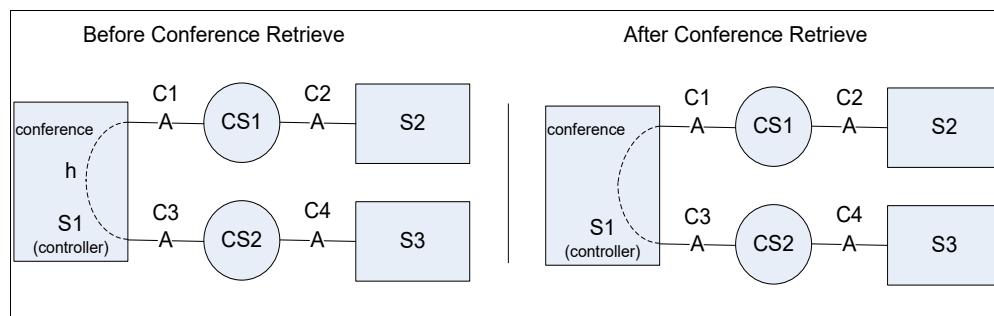


Figure 133 Conference Retrieved Message Flow

Activity	Monitored Subscriber S1	Monitored Subscriber S2	Monitored Subscriber S3	Comments
Conference Retrieve is requested on behalf of device S1.	Conference Retrieved Request (F1) Conference Retrieved Response (F2)			
Indication that the Request has been initiated from this subscriber.	Conference Retrieved Event (F3) Event Response (F4)			

F1 Request

Parameter Name	Parameter Value
requestId	24
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWZhZ0LmNvbTpwYXNzd29yZA==
ConferenceRetrieveRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/conference/talk
method	PUT
version	Not applicable
params	Not applicable



Parameter Name	Parameter Value
subscriberId	subscriberS1@broadsoft.com
payload	Not applicable

F2 Response

Parameter Name	Parameter Value
requestId	24
sessionId	74
statusCode	200
reason	OK
payload	Not applicable

F3 ConferenceRetrievedEvent

Parameter Name	Parameter Value
requestId	9:193
payload	Not applicable
Event	Not applicable
eventID	ced79306-2fcf-4140-a264-8792e526cf5
sequenceNumber	14
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	bbae6a39-cf9f-4420-91a4-8dd734248176
channelId	d7de7b9b-6335-441a-aba6-b2d7dbd4dc74
targetId	subscriberS1@broadsoft.com
EventData	Not applicable
conference	Not applicable
state	Active
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
appearance	2
conferenceParticipantList	Not applicable
conferenceParticipant	Not applicable
callId	callhalf-24131:0
conferenceParticipant	Not applicable
callId	callhalf-24131:1

F4 EventResponse

Parameter Name	Parameter Value
requestId	9:193
payload	Not applicable
EventResponse	Not applicable
eventID	ced79306-2fcf-4140-a264-8792e526cf5
statusCode	200
reason	OK

9.2.6 Conference Add Participant

This scenario illustrates a successful Conference Add Participant Request. Subscriber S4 calls subscriber S1 and is added to the existing conference between S1, S2 and S3. Once the sequence is completed, S1, S2, S3, and S4 are together on the conference.

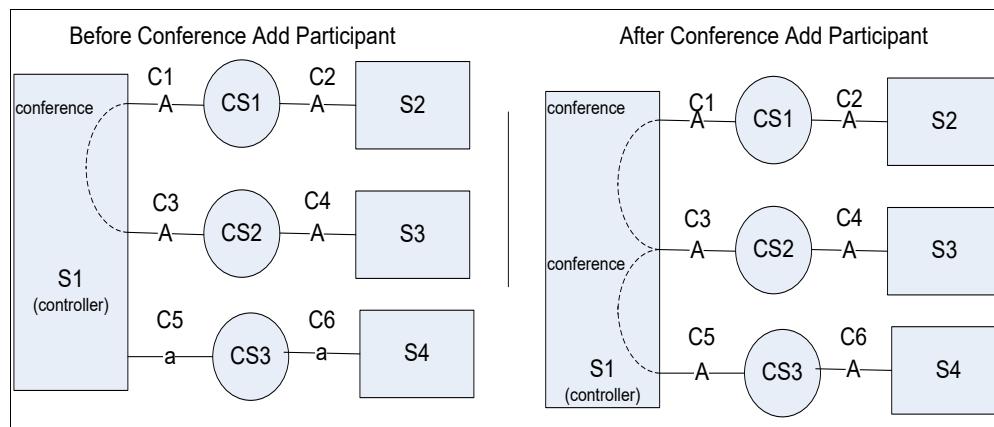


Figure 134 Conference Add Participant Message Flow

Activity	Monitored Subscriber S1	Monitored Subscriber S4	Comments
Conference Add Participant is requested on behalf of S1.	Conference Add Participant Request (F1) Conference Add Participant Response (F2)		Call C5 is an incoming from S1 perspective.
Indication that call C5 is no longer associated with an endpoint.	Call Updated Event (F3) Event Response (F4)		
Indication that Call C6 is now in the Active state.		Call Answered Event (F5) Event Response (F6)	
Indication that the Request has been initiated from this subscriber.	Conference Updated Event (F7) Event Response (F8)		

Activity	Monitored Subscriber S1	Monitored Subscriber S4	Comments
Indication that Call C5 is now in the <i>Active</i> state.	Call Answered Event (F9) Event Response (F10)		

F1 Request

Parameter Name	Parameter Value
requestId	3
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
ConferenceAddParticipantRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/conference/%callId%
method	PUT
version	Not applicable
params	Not applicable
callId	callhalf-24423:8
subscriberId	subscriberS1@broadsoft.com
payload	Not applicable

F2 Response

Parameter Name	Parameter Value
requestId	3
sessionId	94
statusCode	200
reason	OK
payload	Not applicable

F3 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:315
payload	Not applicable
Event	Not applicable
eventID	937b3e0e-0787-4d6f-a060-7a7b253531e9
sequenceNumber	4
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	4fb8ba19-a2bc-4dd2-abc2-3fcfd56c492a4

Parameter Name	Parameter Value
channelId	115a6437-74d8-4348-9145-3dabff3c2b1a
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-24423:8
extTrackingId	140:1
personality	Terminator
state	Alerting
remoteParty	Not applicable
name	subscriberS4FirstName subscriberS4LastName
address	tel:1014
userId	subscriberS4@broadsoft.com
userDN	tel:+15006001014;ext=1014
callType	Group
startTime	1271368559103

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:315
payload	Not applicable
EventResponse	Not applicable
eventId	937b3e0e-0787-4d6f-a060-7a7b253531e9
statusCode	200
reason	OK

F5 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:316
payload	Not applicable
Event	Not applicable
eventId	0745c9ef-b4a8-4eea-ac14-3b1e74bda154
sequenceNumber	3
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	531f906f-171f-4bb9-9338-a6f76d996794
channelId	115a6437-74d8-4348-9145-3dabff3c2b1a

Parameter Name	Parameter Value
targetId	subscriberS4@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-24551:0
extTrackingId	140:1
personality	Originator
state	Active
remoteParty	Not applicable
name	subscriberS1FirstName subscriberS1LastName
address	tel:1011
userId	subscriberS1@broadsoft.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	Not applicable
addressOfRecord	5006001014@broadsoft.com
appearance	1
startTime	1271368559103
answerTime	1271368591930

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:316
payload	Not applicable
EventResponse	Not applicable
eventId	0745c9ef-b4a8-4eea-ac14-3b1e74bda154
statusCode	200
reason	OK

F7 ConferenceUpdatedEvent

Parameter Name	Parameter Value
requestId	0:317
payload	Not applicable
Event	Not applicable
eventId	3dc68d4e-468f-4e9e-8dec-f5d152734905
sequenceNumber	5
userId	adminCTI@broadsoft.com@broadsoft.com

Parameter Name	Parameter Value
externalApplicationId	AppCtlId
subscriptionId	4fb8ba19-a2bc-4dd2-abc2-3fcd56c492a4
channelId	115a6437-74d8-4348-9145-3dabff3c2b1a
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
conference	Not applicable
state	Active
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
appearance	2
conferenceParticipantList	Not applicable
conferenceParticipant	Not applicable
callId	callhalf-24423:0
conferenceParticipant	Not applicable
callId	callhalf-24423:1
conferenceParticipant	Not applicable
callId	callhalf-24423:8

F8 EventResponse

Parameter Name	Parameter Value
requestId	0:317
payload	Not applicable
EventResponse	Not applicable
eventId	3dc68d4e-468f-4e9e-8dec-f5d152734905
statusCode	200
reason	OK

F9 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:318
payload	Not applicable
Event	Not applicable
eventId	3c4d99e3-05c8-41d7-8de1-b4d497cd7bea
sequenceNumber	6
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId

Parameter Name	Parameter Value
subscriptionId	4fb8ba19-a2bc-4dd2-abc2-3fcf56c492a4
channelId	115a6437-74d8-4348-9145-3dabff3c2b1a
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-24423:8
extTrackingId	140:1
personality	Terminator
state	Active
remoteParty	Not applicable
name	subscriberS4FirstName subscriberS4LastName
address	tel:1014
userId	subscriberS4@broadsoft.com
userDN	tel:+15006001014;ext=1014
callType	Group
startTime	1271368559103
answerTime	1271368591930

F10 EventResponse

Parameter Name	Parameter Value
requestId	0:318
payload	Not applicable
EventResponse	Not applicable
eventId	3c4d99e3-05c8-41d7-8de1-b4d497cd7bea
statusCode	200
reason	OK

9.3 Route Point Message Flows (AS only)

The next subsections illustrate the following scenario:

- Call received with treatment and then distributed
- Blind Transfer
- Bounced and Abandoned Call
- Release Call
- Call Updated
- Fail Call
- Call Overflow

- Call Forwarded
- Holiday Policy Applied
- Monitor next Route Point call
- Outgoing Call distributed to agent
- Route Point Failure and recovery

Note that throughout the current section, the RoutePointQueueEntry has been changed to RPQEntry for readability reasons.

9.3.1 Call Received with Treatment and Then Distributed

This scenario illustrates the case where a network party call is added to a Route Point queue and then a treatment is played to the caller. The call is then distributed to agent A1 who answers the call.

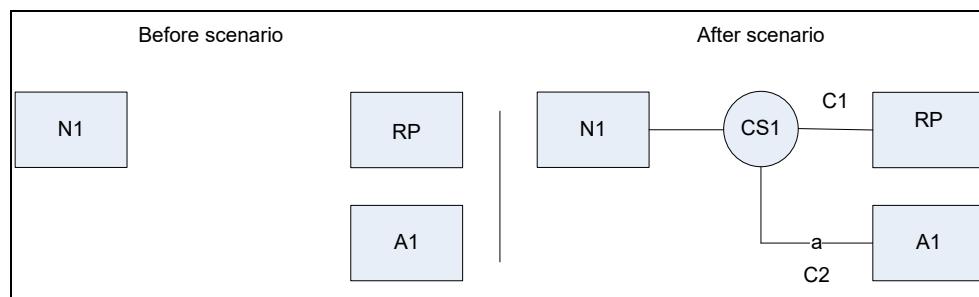


Figure 135 Call Received with Treatment and Then Distributed

Activity	Monitored Route Point	Monitored Agent A1	Comments
An incoming call is received at the Route Point and is screened against the incoming call policy. It is then added to the Route Point queue.	Event: Route Point Call Added Event (F1) Response (F2)		
A Route Point Play Treatment is invoked on the newly added call.	Request: Route Point Play Treatment Request (F3) Response (F4) Event: Route Point Treatment Started (F4)		The Request response contains the playCollectId associated with the operation.
Indication that the treatment is started.	Route Point Treatment Started Event (F5) Response (F6)		
The treatment is completed.	Event: Route Point Treatment Completed Event (F7) Response (F8)		The Play Treatment event specifies that the prompt playback has completed.
A Route Point Play Silence is invoked on the newly added call.	Request: Route Point Play Silence Request (F9) Response (F10) Event: Route Point Silence Started (F8)		The Request response contains the playCollectId associated with the operation.



Activity	Monitored Route Point	Monitored Agent A1	Comments
Indication that the treatment is started.	Event: Route Point Silence Started Event (F11) Response (F12)		
A Route Point Distribute Call is performed on the newly added call to Agent A1. The request specifies to play a whisper message to the agent.	Request: Route Point Distribute Call (F13) Response (F14)		
Automatic ringback is played to caller.	Event: Route Point Silence Completed (F15) Response (F16)		Ringback is automatically played to agent per route point policy. The silence playback is automatically stopped.
The call is offered to the agent. Agent A1 phone begins to ring.	Event: Route Point Call Offered to Agent Event (F17) Response (F18)	Event: Call Received Event (F19) Response (F20)	
The agent A1 answers the call and hears the whisper message.	Event: Route Point Whisper Started Event (F25) Response (F26)	Event: Hook Status (F21) Response (F22) Event: Call Answered Event (F23) Response (F24)	
The whisper message ends.	Route Point Call Answered by Agent Event (F27) Response (F28)		The call is taken out of the Route Point queue. The Communication is established between the agent and the caller.

F1 RoutePointCallAddedEvent

Parameter Name	Parameter Value
requestId	0:2027
payload	Not applicable
Event	Not applicable
eventID	1e080356-2c5a-45b9-89b1-70b9fc1a29ac
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	645aa4dd-5388-472c-8d66-4da9ead72592
channelId	6f1400ca-b86d-4fce-8b91-0aad6e9b7efa
targetId	RoutePoint@broadsoft.com
eventData	Not applicable
queueEntry	Not applicable
callId	callhalf-31157:0
extTrackingId	389:1



Parameter Name	Parameter Value
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271449361084
routePointName	RoutePointName
routePointNumber	tel:+15006001030

F2 EventResponse

Parameter Name	Parameter Value
requestId	0:2027
payload	Not applicable
EventResponse	Not applicable
eventID	1e080356-2c5a-45b9-89b1-70b9fc1a29ac
statusCode	200
reason	OK

F3 Request

Parameter Name	Parameter Value
requestId	3
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
RoutePointPlayTreatmentRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/routepoint/%subscriberId%/calls/%callId%/playtreatment
method	PUT
version	Not applicable
params	Not applicable
callId	callhalf-31157:0
subscriberId	RoutePoint@broadsoft.com
payload	Not applicable
RoutePointPlayTreatment	Not applicable
audioUrlList	Not applicable
uri	http://192.168.13.180/media/en/QueueDefaultEntranceNoEscape.wav

F4 Response

Parameter Name	Parameter Value
requestId	3
sessionId	376
statusCode	200
reason	Not applicable
payload	Not applicable
RoutePointPlayInfo	Not applicable
playCollectId	0

F5 RoutePointTreatmentStartedEvent

Parameter Name	Parameter Value
requestId	0:2028
payload	Not applicable
Event	Not applicable
eventID	68659f75-a31c-4864-b0dd-b8d57ab17dc0
sequenceNumber	3
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	645aa4dd-5388-472c-8d66-4da9ead72592
channelId	6f1400ca-b86d-4cf8-8b91-0aad6e9b7efa
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-31157:0
extTrackingId	389:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271449361084
routePointName	RoutePointName
routePointNumber	tel:+15006001030
playCollectInfo	Not applicable
playCollectId	0
playCollectType	Treatment



F6 EventResponse

Parameter Name	Parameter Value
requestId	0:2028
payload	Not applicable
EventResponse	Not applicable
eventID	68659f75-a31c-4864-b0dd-b8d57ab17dc0
statusCode	200
reason	OK

F7 RoutePointTreatmentCompletedEvent

Parameter Name	Parameter Value
requestId	0:2029
payload	Not applicable
Event	Not applicable
eventID	51c08374-4e68-402c-8c64-178d42cd5e51
sequenceNumber	4
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	645aa4dd-5388-472c-8d66-4da9ead72592
channelId	6f1400ca-b86d-4cfe-8b91-0aad6e9b7efa
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-31157:0
extTrackingId	389:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271449361084
routePointName	RoutePointName
routePointNumber	tel:+15006001030
playCollectResult	Not applicable
playCollectId	0
completionReason	EOF



F8 EventResponse

Parameter Name	Parameter Value
requestId	0:2029
payload	Not applicable
EventResponse	Not applicable
eventID	51c08374-4e68-402c-8c64-178d42cd5e51
statusCode	200
reason	OK

F9 Request

Parameter Name	Parameter Value
requestId	4
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
RoutePointPlay SilenceRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/routepoint/%subscriberId%/calls/%callId%/playsilence
method	PUT
version	Not applicable
params	Not applicable
callId	callhalf-31157:0
subscriberId	RoutePoint@broadsoft.com
payload	Not applicable
RoutePointPlay Silence	Not applicable
duration	10

F10 Response

Parameter Name	Parameter Value
requestId	4
sessionId	378
statusCode	200
reason	Not applicable
payload	Not applicable
RoutePointPlayInfo	Not applicable
playCollectId	1



F11 RoutePointSilenceStartedEvent

Parameter Name	Parameter Value
requestId	0:2030
payload	Not applicable
Event	Not applicable
eventID	fad528a2-1c8c-4785-b32a-01b2915ea91b
sequenceNumber	5
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	645aa4dd-5388-472c-8d66-4da9ead72592
channelId	6f1400ca-b86d-4cfe-8b91-0aad6e9b7efa
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-31157:0
extTrackingId	389:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271449361084
routePointName	RoutePointName
routePointNumber	tel:+15006001030
playCollectInfo	Not applicable
playCollectId	1
playCollectType	Silence

F12 EventResponse

Parameter Name	Parameter Value
requestId	0:2030
payload	Not applicable
EventResponse	Not applicable
eventID	fad528a2-1c8c-4785-b32a-01b2915ea91b
statusCode	200
reason	OK



F13 Request

Parameter Name	Parameter Value
requestId	5
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
RoutePointDistribute CallRequest	Not applicable
uri	/com.broadsoft.xsi- actions/v2.0/routePoint/%subscriberId%/calls/%callId%/distribute
method	PUT
version	Not applicable
params	Not applicable
callId	callhalf-31157:0
subscriberId	RoutePoint@broadsoft.com
payload	Not applicable
RoutePointDistribute	Not applicable
agentId	agentA1@broadsoft.com
callingLineIdNumber	tel:5006001030
callingLineIdName	RoutePoint
timeInQueue	1
numberOfCallsIn Queue	1
longestWaitTime	1
whisperMessage AudioUrl	http://192.168.13.180/media/en/whisper.wav

F14 Response

Parameter Name	Parameter Value
requestId	5
sessionId	380
statusCode	200
reason	Not applicable
payload	Not applicable

F15 RoutePointSilenceCompletedEvent

Parameter Name	Parameter Value
requestId	0:2031
payload	Not applicable
Event	Not applicable

Parameter Name	Parameter Value
eventID	78fe6ed0-f32a-4952-bf66-1cf54eb475b5
sequenceNumber	6
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	645aa4dd-5388-472c-8d66-4da9ead72592
channelId	6f1400ca-b86d-4cfe-8b91-0aad6e9b7efa
targetId	RoutePoint@broadsoft.com
eventData	Not applicable
queueEntry	Not applicable
callId	callhalf-31157:0
extTrackingId	389:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271449361084
routePointName	RoutePointName
routePointNumber	tel:+15006001030
playCollectResult	Not applicable
playCollectId	1
completionReason	EOF

F16 EventResponse

Parameter Name	Parameter Value
requestId	0:2031
payload	Not applicable
EventResponse	Not applicable
eventID	78fe6ed0-f32a-4952-bf66-1cf54eb475b5
statusCode	200
reason	OK

F17 RoutePointCallOfferedToAgentEvent

Parameter Name	Parameter Value
requestId	0:2036
payload	Not applicable
Event	Not applicable



Parameter Name	Parameter Value
eventID	cb322733-3243-402d-bc0c-5ca8fafdfba7
sequenceNumber	7
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	645aa4dd-5388-472c-8d66-4da9ead72592
channelId	6f1400ca-b86d-4cfe-8b91-0aad6e9b7efa
targetId	RoutePoint@broadsoft.com
eventData	Not applicable
queueEntry	Not applicable
callId	callhalf-31157:0
extTrackingId	389:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271449361084
routePointName	RoutePointName
routePointNumber	tel:+15006001030

F18 EventResponse

Parameter Name	Parameter Value
requestId	0:2036
payload	Not applicable
EventResponse	Not applicable
eventID	cb322733-3243-402d-bc0c-5ca8fafdfba7
statusCode	200
reason	OK

F19 CallReceivedEvent

Parameter Name	Parameter Value
requestId	0:2037
payload	Not applicable
Event	Not applicable
eventID	9ca63f3e-3f52-4573-874f-9db71f4e9477
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com



Parameter Name	Parameter Value
externalApplicationId	AppCtlId
subscriptionId	1715e3bd-863a-42a2-aa59-88d8db38bbe7
channelId	6f1400ca-b86d-4cf8-8b91-0aad6e9b7efa
targetId	agentA1@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-31175:0
extTrackingId	389:1
personality	Terminator
state	Alerting
remoteParty	Not applicable
name	RoutePointFirstName RoutePointLastName – NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
startTime	1271449535817
acdCallInfo	Not applicable
acdUserId	RoutePoint@broadsoft.com
acdName	RoutePointName
acdNumber	tel:+15006001030
numCallsInQueue	0
waitTime	1
longestWaitTime	1
callingPartyInfo	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network

F20 EventResponse

Parameter Name	Parameter Value
requestId	0:2037
payload	Not applicable
EventResponse	Not applicable
eventID	9ca63f3e-3f52-4573-874f-9db71f4e9477
statusCode	200
reason	OK



F21 HookStatusEvent

Parameter Name	Parameter Value
RequestId	0:1973
Payload	Not applicable
Event	Not applicable
EventID	68ea76c2-660e-43b1-b1ff-f32931db2854
sequenceNumber	3
UserId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	1715e3bd-863a-42a2-aa59-88d8db38bbe7
channelId	6f1400ca-b86d-4cfe-8b91-0aad6e9b7efa
Targeted	agentA1@broadsoft.com
eventData	Not applicable
hookStatus	Off-Hook

F22 EventResponse

Parameter Name	Parameter Value
requestId	0:1973
Payload	Not applicable
EventResponse	Not applicable
eventid	68ea76c2-660e-43b1-b1ff-f32931db2854
statusCode	200
reason	OK

F23 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:2038
payload	Not applicable
Event	Not applicable
eventID	90f2defd-0f7f-48f7-8314-e249f4add5a7
sequenceNumber	4
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	1715e3bd-863a-42a2-aa59-88d8db38bbe7
channelId	6f1400ca-b86d-4cfe-8b91-0aad6e9b7efa
targetId	agentA1@broadsoft.com
eventData	Not applicable



Parameter Name	Parameter Value
call	Not applicable
callId	callhalf-31175:0
extTrackingId	389:1
personality	Terminator
state	Active
remoteParty	Not applicable
name	RoutePointFirstName RoutePointLastName - NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
endpoint	Not applicable
addressOfRecord	5006001020@broadsoft.com
appearance	1
startTime	1271449535817
answerTime	1271449544204
acdCallInfo	Not applicable
acdUserId	RoutePoint@broadsoft.com
acdName	RoutePointName
acdNumber	tel:+15006001030
numCallsInQueue	0
waitTime	1
longestWaitTime	1
callingPartyInfo	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network

F24 EventResponse

Parameter Name	Parameter Value
requestId	0:2038
payload	Not applicable
EventResponse	Not applicable
eventID	90f2defd-0f7f-48f7-8314-e249f4add5a7
statusCode	200
reason	OK



F25 RoutePointWhisperStartedEvent

Parameter Name	Parameter Value
requestId	0:2039
payload	Not applicable
Event	Not applicable
eventID	64fdc43f-6212-4a82-82b7-9f2de10a5b99
sequenceNumber	8
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	645aa4dd-5388-472c-8d66-4da9ead72592
channelId	6f1400ca-b86d-4cfe-8b91-0aad6e9b7efa
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-31157:0
extTrackingId	389:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271449361084
routePointName	RoutePointName
routePointNumber	tel:+15006001030
answeringUserId	agentA1@broadsoft.com
answeringCallId	callhalf-31175:0

F26 EventResponse

Parameter Name	Parameter Value
requestId	0:2039
payload	Not applicable
EventResponse	Not applicable
eventID	64fdc43f-6212-4a82-82b7-9f2de10a5b99
statusCode	200
reason	OK



F27 RoutePointCallAnsweredByAgentEvent

Parameter Name	Parameter Value
requestId	0:2040
payload	Not applicable
Event	Not applicable
eventID	e41f8466-13a7-4f8f-ad37-ceea24351948
sequenceNumber	9
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	645aa4dd-5388-472c-8d66-4da9ead72592
channelId	6f1400ca-b86d-4cfe-8b91-0aad6e9b7efa
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-31157:0
extTrackingId	389:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271449361084
removeTime	1271449549576
routePointName	RoutePointName
routePointNumber	tel:+15006001030
answeringUserId	agentA1@broadsoft.com
answeringCallId	callhalf-31175:0

F28 EventResponse

Parameter Name	Parameter Value
requestId	0:2040
payload	Not applicable
EventResponse	Not applicable
eventID	e41f8466-13a7-4f8f-ad37-ceea24351948
statusCode	200
reason	OK

9.3.2 Transfer

This scenario illustrates a successful Route Point Blind Transfer. Network party N1 is connected to the Route Point RP via call C1 when the remote application initiates a blind transfer to subscriber S1. In this scenario, subscriber S1 is available and valid, and the call is answered by subscriber S1.

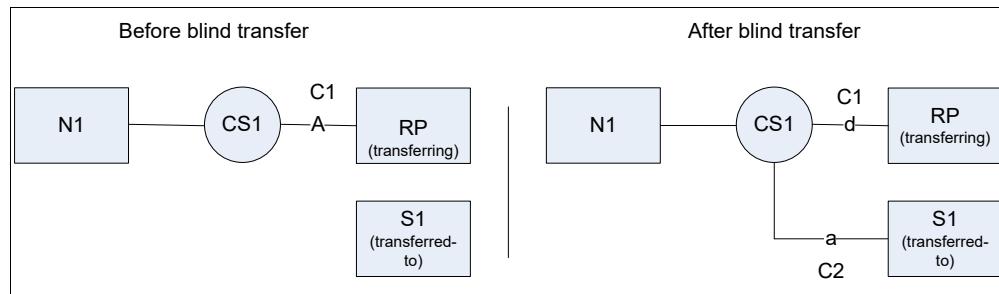


Figure 136 Transfer Message Flow

Activity	Monitored RP	Monitored Subscriber S1	Comments
A Transfer Request to subscriber S1 is invoked on behalf of RP.	Route Point Blind Transfer Request (F1) Route Point Blind Transfer Response (F2)		
Indication that the Request has been initiated from this subscriber.	Route Point Call Transferred Event (F3) Event Response (F4)		The Call Transferred event specifies that the call is now in <i>Alerting</i> state and remoteParty is now S3.
Subscriber S1 is alerted.		Call Received Event (F5) Event Response (F6)	The Call Received event contains detail on the redirection that was performed.

F1 Request

Parameter Name	Parameter Value
requestId	9
sessionId	MySessionId
credentials	YWRtaW5DVElAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
RoutePointBlindTransfer Request	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/routepoint/%routepointid%/calls/%callid%/BlindTransfer?address=%address%
method	PUT
version	Not applicable
params	Not applicable
address	tel:1011@broadsoft.com



Parameter Name	Parameter Value
routepointid	RoutePoint@broadsoft.com
callid	callhalf-29295:0
payload	Not applicable

F2 Response

Parameter Name	Parameter Value
requestId	9
sessionId	233
statusCode	200
reason	OK
payload	Not applicable

F3 RoutePointCallTransferredEvent

Parameter Name	Parameter Value
requestId	0:1671
payload	Not applicable
Event	Not applicable
eventID	fb9a4fc1-d056-442c-b244-45cfba9f618b
sequenceNumber	3
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	16fb2483-7651-4067-b0a4-807b5f854d6d
channelId	5bdbfbf2-038f-4c64-9032-672ee914f69b
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-29713:0
extTrackingId	252:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271439887783
removeTime	1271439923771
routePointName	RoutePointName
routePointNumber	tel:+15006001030

Parameter Name	Parameter Value
redirect	Not applicable
address	tel:5006001011
reason	deflection
redirectTime	1271439923771

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:1671
payload	Not applicable
EventResponse	Not applicable
eventID	fb9a4fc1-d056-442c-b244-45cfba9f618b
statusCode	200
reason	OK

F5 CallReceivedEvent

Parameter Name	Parameter Value
requestId	0:1672
payload	Not applicable
Event	Not applicable
eventID	701f9d1a-fc7a-44b0-8903-58cf7be5fc9
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	86fff0ea-b878-4225-8f86-cd53f3d8b6ab
channelId	5bdbfbf2-038f-4c64-9032-672ee914f69b
targetId	subscriberS1@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-29729:0
extTrackingId	252:1
personality	Terminator
state	Alerting
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network

Parameter Name	Parameter Value
startTime	1271439923772

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:1672
payload	Not applicable
EventResponse	Not applicable
eventID	701f9d1a-fc7a-44b0-8903-58cf7be5fc9
statusCode	200
reason	OK

9.3.3 Bounced and Abandoned Call

This scenario illustrates a case where a call offered to an agent is bounced because an action is performed by the remote application. The call is eventually dropped by the caller.

In this scenario, network party N1 is initially connected to the Route Point RP via call C1. Call C1 is maintained in the Route Point queue.

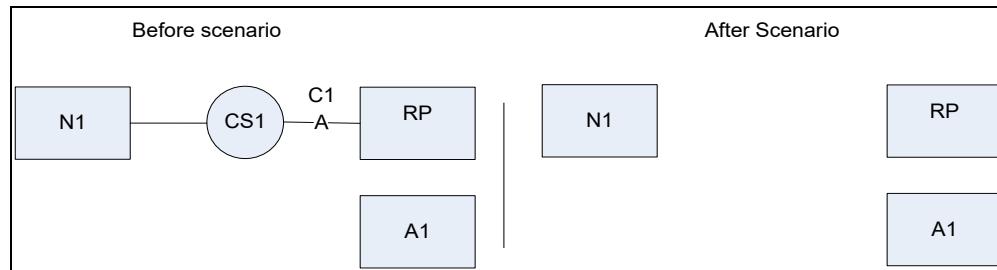


Figure 137 Bounced and Abandoned Call Message Flow

Activity	Monitored RP	Monitored Agent A1	Comments
A Route Point Distribute Call is performed on the queued call C1 to Agent A1	Request: Route Point Distribute Call Request (F1) Route Point Distribute Call Response (F2)		
The call is offered to the agent. Agent A1 phone begins to ring.	Event: Route Point Call Offered to Agent Event (F2) Event Response (F3)	Event: Call Received Event (F5) Event Response (F6)	
The Remote application bounces the call using the Bounce Call Request.	Request: Bounce Call Request (F7) Bounce Call Response (F8) Event: Route Point Call Bounced (F9)	Event: Call Released. (F9)	The RP bounced the call.



Activity	Monitored RP	Monitored Agent A1	Comments
Indication that the call is bounced and released at the agent level.	Route Point Call Bounced Event (F9) Event Response (F10)	Call Released Event (F11) Event Response (F12)	
MOH is invoked on call C1.	Request: Route Point Play MOH Request (F13) Response (F16)		The Request response contains the playCollectId associated with the operation.
Indication that the MOH has been started.	Event: Route Point Play MOH Started Event (F14) Event Response (F15)		The ringback is stopped when a call is bounced.
Indication that the MOH has completed.	Route Point Play MOH Completed Event (F17) Event Response (F18)		
Caller N1 hangs up the phone.	Event: Route Point Call Abandoned (F19) Event Response (F20)		

F1 Request

Parameter Name	Parameter Value
requestId	14
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
RoutePointDistribute CallRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/routepoint/%subscriberId%/calls/%callId%/distribute
method	PUT
version	Not applicable
params	Not applicable
callId	callhalf-30337:0
subscriberId	RoutePoint@broadsoft.com
payload	Not applicable
RoutePointDistribute	Not applicable
agentId	agentA1@broadsoft.com
callingLineIdNumber	tel:5006001011@broadsoft.com
callingLineIdName	RoutePoint
timeInQueue	1



Parameter Name	Parameter Value
numberOfCallsInQueue	1
longestWaitTime	1

F2 Response

Parameter Name	Parameter Value
requestId	14
sessionId	292
statusCode	200
reason	Not applicable
payload	Not applicable

F3 RoutePointCallOfferedToAgentEvent

Parameter Name	Parameter Value
requestId	1:1833
payload	Not applicable
Event	Not applicable
eventID	3a405d0d-3442-47f6-a935-f13069f30d88
sequenceNumber	14
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	51cd2322-df6b-4249-b33d-a9927fe984d7
channelId	cdc704c3-3f89-4b3b-bccc-10ba0abe29ad
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-30337:0
extTrackingId	312:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271443969817
routePointName	RoutePointName
routePointNumber	tel:+15006001030



F4 EventResponse

Parameter Name	Parameter Value
requestId	1:1833
payload	Not applicable
EventResponse	Not applicable
eventID	3a405d0d-3442-47f6-a935-f13069f30d88
statusCode	200
reason	OK

F5 CallReceivedEvent

Parameter Name	Parameter Value
requestId	1:1834
payload	Not applicable
Event	Not applicable
eventID	6cb810ff-d7d2-45c7-b4a3-b441c49559ea
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	6fa2b723-fdf7-4435-acce-2f8fb803f33b
channelId	cdc704c3-3f89-4b3b-bccc-10ba0abe29ad
targetId	agentA1@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-30341:0
extTrackingId	312:1
personality	Terminator
state	Alerting
remoteParty	Not applicable
name	RoutePointFirstName RoutePointLastName - NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
startTime	1271444044672
acdCallInfo	Not applicable
acdUserId	RoutePoint@broadsoft.com
acdName	RoutePointName
acdNumber	tel:+15006001030



Parameter Name	Parameter Value
numCallsInQueue	0
waitTime	1
longestWaitTime	1
callingPartyInfo	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network

F6 EventResponse

Parameter Name	Parameter Value
requestId	1:1834
payload	Not applicable
EventResponse	Not applicable
eventID	6cb810ff-d7d2-45c7-b4a3-b441c49559ea
statusCode	200
reason	OK

F7 Request

Parameter Name	Parameter Value
requestId	15
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
RoutePointBounce CallRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/routepoint/%subscriberId%/calls/%callId%/bounce
method	PUT
version	Not applicable
params	Not applicable
callId	callhalf-30337:0
subscriberId	RoutePoint@broadsoft.com
payload	Not applicable

F8 Response

Parameter Name	Parameter Value
requestId	15
sessionId	293
statusCode	200



Parameter Name	Parameter Value
reason	Not applicable
payload	Not applicable

F9 RoutePointCallBouncedEvent

Parameter Name	Parameter Value
requestId	1:1835
payload	Not applicable
Event	Not applicable
eventID	eba9df6a-2ca1-49a6-b5ae-19b614ce104c
sequenceNumber	15
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	51cd2322-df6b-4249-b33d-a9927fe984d7
channelId	cdc704c3-3f89-4b3b-bccc-10ba0abe29ad
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-30337:0
extTrackingId	312:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271443969817
routePointName	RoutePointName
routePointNumber	tel:+15006001030

F10 EventResponse

Parameter Name	Parameter Value
requestId	1:1835
payload	Not applicable
EventResponse	Not applicable
eventID	eba9df6a-2ca1-49a6-b5ae-19b614ce104c
statusCode	200
reason	OK



F11 CallReleasedEvent

Parameter Name	Parameter Value
requestId	1:1836
payload	Not applicable
Event	Not applicable
eventID	f4df09db-01f0-42db-8fe4-84103ddd8a0d
sequenceNumber	3
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	6fa2b723-fdf7-4435-acce-2f8fb803f33b
channelId	cdc704c3-3f89-4b3b-bccc-10ba0abe29ad
targetId	agentA1@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-30341:0
extTrackingId	312:1
personality	Terminator
state	Released
releasingParty	remoteRelease
remoteParty	Not applicable
name	RoutePointFirstName RoutePointLastName - NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
endpoint	Not applicable
addressOfRecord	5006001020@broadsoft.com
startTime	1271444044672
releaseTime	1271444063505
acdCallInfo	Not applicable
acdUserId	RoutePoint@broadsoft.com
acdName	RoutePointName
acdNumber	tel:+15006001030
numCallsInQueue	0
waitTime	1
longestWaitTime	1
callingPartyInfo	Not applicable
name	NetworkN1FirstName NetworkN1LastName



Parameter Name	Parameter Value
address	tel:+15007003333
callType	Network

F12 EventResponse

Parameter Name	Parameter Value
requestId	1:1836
payload	Not applicable
EventResponse	Not applicable
eventID	f4df09db-01f0-42db-8fe4-84103ddd8a0d
statusCode	200
reason	OK

F13 Request

Parameter Name	Parameter Value
requestId	17
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
RoutePointPlayMOHRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/routepoint/%subscriberId%/calls/%callId%/playmusiconhold
method	PUT
version	Not applicable
params	Not applicable
callId	callhalf-30337:0
subscriberId	RoutePoint@broadsoft.com
payload	Not applicable
RoutePointPlayMusicOnHold	Not applicable
duration	10

F14 RoutePointMOHStartedEvent

Parameter Name	Parameter Value
requestId	1:1838
payload	Not applicable
Event	Not applicable
eventID	24b2b5e1-3d0c-413a-8600-805cf75738b3
sequenceNumber	16



Parameter Name	Parameter Value
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	51cd2322-df6b-4249-b33d-a9927fe984d7
channelId	cdc704c3-3f89-4b3b-bccc-10ba0abe29ad
targetId	RoutePoint@broadsoft.com
eventData	Not applicable
queueEntry	Not applicable
callId	callhalf-30337:0
extTrackingId	312:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271443969817
routePointName	RoutePointName
routePointNumber	tel:+15006001030
playCollectInfo	Not applicable
playCollectId	0
playCollectType	MOH

F15 EventResponse

Parameter Name	Parameter Value
requestId	1:1838
payload	Not applicable
EventResponse	Not applicable
eventId	24b2b5e1-3d0c-413a-8600-805cf75738b3
statusCode	200
reason	OK

F16 Response

Parameter Name	Parameter Value
requestId	17
sessionId	295
statusCode	200
reason	Not applicable
payload	Not applicable



Parameter Name	Parameter Value
RoutePointPlayInfo	Not applicable
playCollectId	0

F17 RoutePointMOHCompletedEvent

Parameter Name	Parameter Value
requestId	1:1840
payload	Not applicable
Event	Not applicable
eventID	d75ee47f-ec37-4362-98eb-20ee0a305140
sequenceNumber	17
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	51cd2322-df6b-4249-b33d-a9927fe984d7
channelId	cdc704c3-3f89-4b3b-bccc-10ba0abe29ad
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-30337:0
extTrackingId	312:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271443969817
routePointName	RoutePointName
routePointNumber	tel:+15006001030
playCollectResult	Not applicable
playCollectId	0
completionReason	EOF

F18 EventResponse

Parameter Name	Parameter Value
requestId	1:1840
payload	Not applicable
EventResponse	Not applicable
eventID	d75ee47f-ec37-4362-98eb-20ee0a305140



Parameter Name	Parameter Value
statusCode	200
reason	OK

F19 RoutePointCallAbandonedEvent

Parameter Name	Parameter Value
requestId	1:1841
payload	Not applicable
Event	Not applicable
eventID	41dd7c1d-6369-448b-b36c-d166990ac26c
sequenceNumber	18
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	51cd2322-df6b-4249-b33d-a9927fe984d7
channelId	cdc704c3-3f89-4b3b-bccc-10ba0abe29ad
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-30337:0
extTrackingId	312:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271443969817
removeTime	1271444140325
routePointName	RoutePointName
routePointNumber	tel:+15006001030

F20 EventResponse

Parameter Name	Parameter Value
requestId	1:1841
payload	Not applicable
EventResponse	Not applicable
eventID	41dd7c1d-6369-448b-b36c-d166990ac26c
statusCode	200
reason	OK

9.3.4 Release Call

This scenario illustrates a case where a call is released through a Route Point Release Call while being offered to an agent.

In this scenario, network party N1 is initially connected to the Route Point RP via call C1. Call C1 is maintained in the Route Point queue and offered to agent A1.

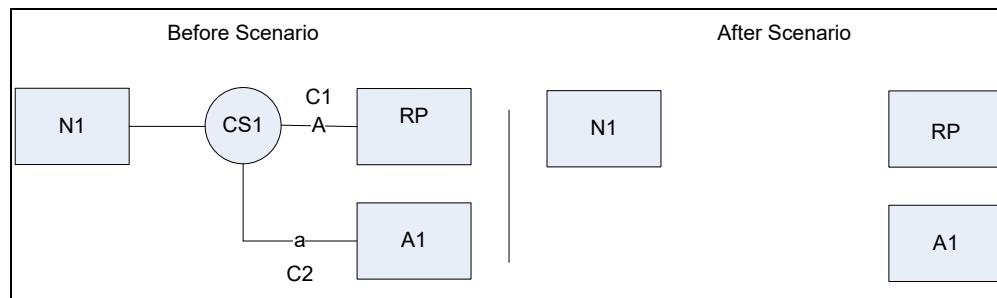


Figure 138 Release Call Message Flow

Activity	Monitored RP	Monitored Agent A1	Comments
A Route Point Release Call is performed on the queued call C1 offered to Agent A1.	Route Point Release Call Request (F1) Response (F2)		
	Route Point Call Released Event (F3) Response (F4)	Call Released Event (F5) Response (F6)	

F1 Request

Parameter Name	Parameter Value
requestId	8
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzbZ0LmNvbTpwYXNzd29yZA==
RoutePointReleaseCall Request	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/routepoint/%subscriberId%/calls/%callId%
method	DELETE
version	Not applicable
params	Not applicable
callId	callhalf-30871:0
subscriberId	RoutePoint@broadsoft.com
payload	Not applicable

F2 Response

Parameter Name	Parameter Value
requestId	8



Parameter Name	Parameter Value
sessionId	326
statusCode	200
reason	Not applicable
payload	Not applicable

F3 RoutePointCallReleasedEvent

Parameter Name	Parameter Value
requestId	0:1943
payload	Not applicable
Event	Not applicable
eventID	48fc51b0-3332-4e00-a162-2a9c1960b8ae
sequenceNumber	7
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	33df61df-ae00-4494-96c5-72f5faa07653
channelId	e4d4c6c1-611b-4d4a-a00f-127fbca994bf
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-30871:0
extTrackingId	360:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271446626899
removeTime	1271446701922
routePointName	RoutePointName
routePointNumber	tel:+15006001030

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:1943
payload	Not applicable
EventResponse	Not applicable
eventID	48fc51b0-3332-4e00-a162-2a9c1960b8ae



Parameter Name	Parameter Value
statusCode	200
reason	OK

F5 CallReleasedEvent

Parameter Name	Parameter Value
requestId	0:1944
payload	Not applicable
Event	Not applicable
eventID	cae091ea-3601-47a4-9e37-574dd180149b
sequenceNumber	6
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	a0fdf2b5-964e-4606-92f9-e0ad1b2af054
channelId	e4d4c6c1-611b-4d4a-a00f-127fbca994bf
targetId	agentA1@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-30879:0
extTrackingId	360:1
personality	Terminator
state	Released
releasingParty	remoteRelease
remoteParty	Not applicable
name	RoutePointFirstName RoutePointLastName NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
endpoint	Not applicable
addressOfRecord	5006001020@broadsoft.com
startTime	1271446695742
releaseTime	1271446701925
acdCallInfo	Not applicable
acdUserId	RoutePoint@broadsoft.com
acdName	RoutePointName
acdNumber	tel:+15006001030
numCallsInQueue	0

Parameter Name	Parameter Value
waitTime	1
longestWaitTime	1
callingPartyInfo	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:1944
payload	Not applicable
EventResponse	Not applicable
eventID	cae091ea-3601-47a4-9e37-574dd180149b
statusCode	200
reason	OK

9.3.5 Call Updated

This scenario illustrates the case where a Call Updated event is generated. Network party N1 is connected to the Route Point RP via call C1 when the N1 transfer the call to destination N2.

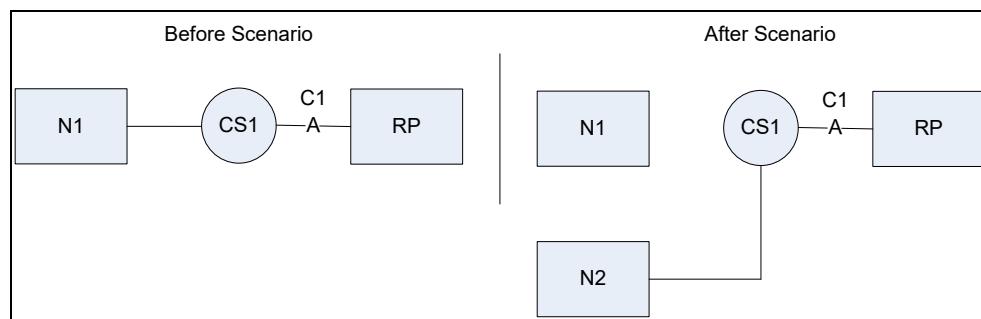


Figure 139 Call Updated Message Flow

Activity	Monitored RP	Comments
A Transfer to Network Party N2 is performed by N1.		
Indication that the remote party has changed.	Route Point Call Updated Event (F1) Response (F2)	The Call Updated event specifies that remoteParty is now N2.

F1 RoutePointCallUpdatedEvent

Parameter Name	Parameter Value
requestId	30:1688



Parameter Name	Parameter Value
payload	Not applicable
Event	Not applicable
eventID	7adb60f5-da38-42ba-b2a5-fbb609289ff6
sequenceNumber	5
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	55c1b1bf-93ac-4bb7-a18e-2e2d31e57e82
channelId	4bfbfd1-e992-49c2-ad2d-764c282b7db8
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-77621:0
extTrackingId	175:1
remoteParty	Not applicable
name	NetworkN2FirstName NetworkN2LastName
address	tel:+15007003334
callType	Network
addTime	1271879264315
routePointName	RoutePointName
routePointNumber	tel:+15006001030
reason	Call Transferred

F2 EventResponse

Parameter Name	Parameter Value
requestId	30:1688
payload	Not applicable
EventResponse	Not applicable
eventID	7adb60f5-da38-42ba-b2a5-fbb609289ff6
statusCode	200
reason	OK

9.3.6 Fail Call

This scenario illustrates a case where a call is failed through a Route Point Fail Call while being offered to an agent.

In this scenario, network party N1 is initially connected to the Route Point RP via call C1. Call C1 is maintained in the Route Point queue and offered to agent A1. The failed destination is configured to be an ACD.

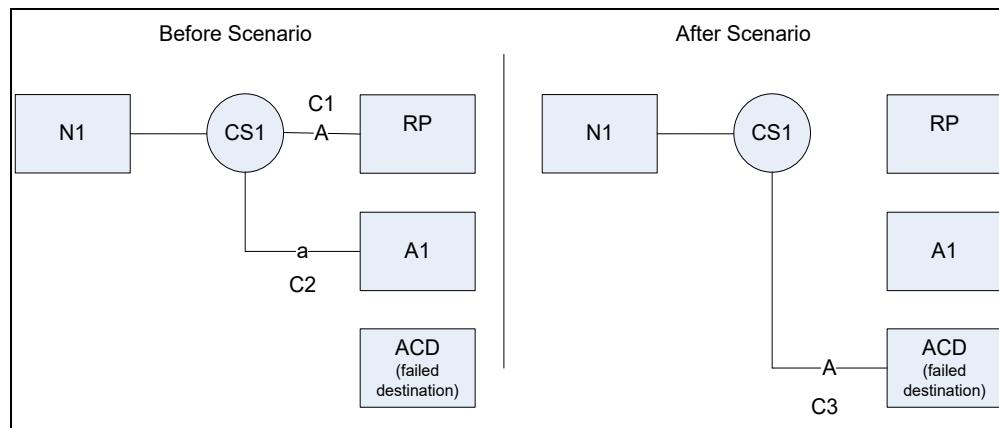


Figure 140 Fail Call Message Flow

Activity	Monitored RP	Monitored ACD	Comments
Agent 1 receives a call from the RP.			Agent 1 has an alerting call.
A Route Point Fail Call is performed on the queued call C1 offered to Agent A1.	Route Point Fail Call Request (F1) Response (F2)		
	Route Point Call Failed Event (F3) Response (F4)		
The Call is added to the queue at destination ACD.		ACD Call Added Event (F5) Response (F6)	

F1 Request

Parameter Name	Parameter Value
requestId	4
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzbZ0LmNvbTpwYXNzd29yZA==
RoutePointFailCall Request	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/routePoint/%subscriberId%/calls/%callId%/fail
method	PUT
version	Not applicable



Parameter Name	Parameter Value
params	Not applicable
callId	callhalf-31515:0
subscriberId	RoutePoint@broadsoft.com
payload	Not applicable

F2 Response

Parameter Name	Parameter Value
requestId	4
sessionId	411
statusCode	200
reason	OK
payload	Not applicable

F3 RoutePointCallFailedEvent

Parameter Name	Parameter Value
requestId	1:2093
payload	Not applicable
Event	Not applicable
eventID	40ecabfd-e007-4599-8c2e-decd1fffd651
sequenceNumber	3
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	42e520bf-51f1-4c89-9a35-851009bcc847
channelId	c35e8826-85b6-4088-abd2-2ca7716943ed
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-31515:0
extTrackingId	417:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271451627658
removeTime	1271451651839
routePointName	RoutePointName



Parameter Name	Parameter Value
routePointNumber	tel:+15006001030
failureReason	CallFailureRequested

F4 EventResponse

Parameter Name	Parameter Value
requestId	1:2093
payload	Not applicable
EventResponse	Not applicable
eventID	40ecabfd-e007-4599-8c2e-decd1ffd651
statusCode	200
reason	OK

F5 ACDCallAddedEvent

Parameter Name	Parameter Value
requestId	1:2094
payload	Not applicable
Event	Not applicable
eventID	79c3babf-022d-45bf-86e3-75226357e32f
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	e6560b3e-f8b5-44a9-b1a1-6632a6be29da
channelId	c35e8826-85b6-4088-abd2-2ca7716943ed
targetId	ACD_CTL1@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-31519:0
extTrackingId	417:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271451651905
acdName	ACD_CTL1_Name
acdNumber	tel:+15006001025
acdPriority	0-Highest

Parameter Name	Parameter Value
addTimeInPriorityBucket	1271451651905
position	1
F6 EventResponse	
Parameter Name	Parameter Value
requestId	1:2094
payload	Not applicable
EventResponse	Not applicable
eventID	79c3babf-022d-45bf-86e3-75226357e32f
statusCode	200
reason	OK

9.3.7 Call Overflow

This scenario illustrates a case where a call has been maintained in the queue for longer than a specified configurable value.

In this scenario, network party N1 is initially connected to the Route Point RP via call C1. Call C1 is maintained in the Route Point queue. The overflow policy is configured to send the call to the ACD.

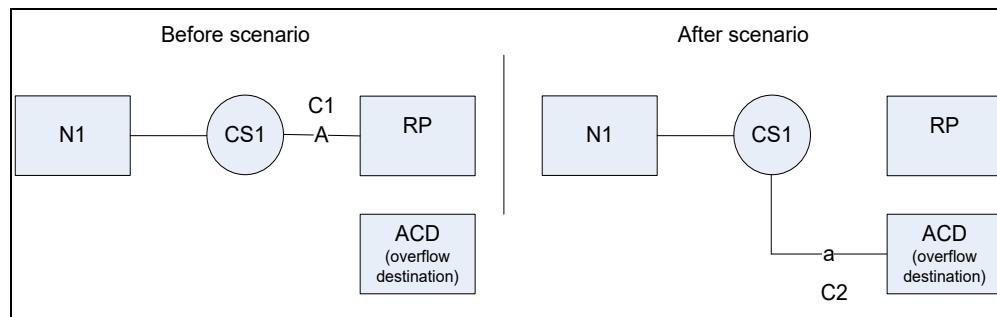


Figure 141 Call Overflow Message Flow

Activity	Monitored RP	Monitored ACD	Comments
The call has been in the queue for longer than the value specified by the overflow policy. A ringing tone is played to the caller as per the overflow policy.	Route Point Call Overflowed Event (F1) Event Response (F2)		
Indication that the call is added to the ACD queue.		ACD Call Added Event (F3) Event Response (F4)	



F1 RoutePointCallOverflowedEvent

Parameter Name	Parameter Value
requestId	1:2126
payload	Not applicable
Event	Not applicable
eventID	5539b571-3d2f-49a9-b555-8f7f91904aad
sequenceNumber	7
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	42e520bf-51f1-4c89-9a35-851009bcc847
channelId	c35e8826-85b6-4088-abd2-2ca7716943ed
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-31629:0
extTrackingId	427:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271452711590
removeTime	1271452721598
routePointName	RoutePointName
routePointNumber	tel:+15006001030
overFlowReason	time
redirect	Not applicable
address	tel:1025
reason	deflection
redirectTime	1271452721616

F2 EventResponse

Parameter Name	Parameter Value
requestId	1:2126
payload	Not applicable
EventResponse	Not applicable
eventID	5539b571-3d2f-49a9-b555-8f7f91904aad
statusCode	200



Parameter Name	Parameter Value
reason	OK

F3 ACDCallAddedEvent

Parameter Name	Parameter Value
requestId	1:2127
payload	Not applicable
Event	Not applicable
eventID	d9dcebee-a536-4fcc-9a46-6a4226e942ea
sequenceNumber	9
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	e6560b3e-f8b5-44a9-b1a1-6632a6be29da
channelId	c35e8826-85b6-4088-abd2-2ca7716943ed
targetId	ACD_CTI1@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-31633:0
extTrackingId	427:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271452721686
acdName	ACD_CTI1_Name
acdNumber	tel:+15006001025
acdPriority	0-Highest
addTimeInPriority Bucket	1271452721686
position	1

F4 EventResponse

Parameter Name	Parameter Value
requestId	1:2127
payload	Not applicable
EventResponse	Not applicable
eventID	d9dcebee-a536-4fcc-9a46-6a4226e942ea
statusCode	200

Parameter Name	Parameter Value
reason	OK

9.3.8 Call Forwarded

This scenario illustrates a case where a call is forwarded because of the Forced Forwarding policy. In this scenario, network party N1 initiates a call toward the Route Point RP for which the Forced Forwarding policy is enabled.

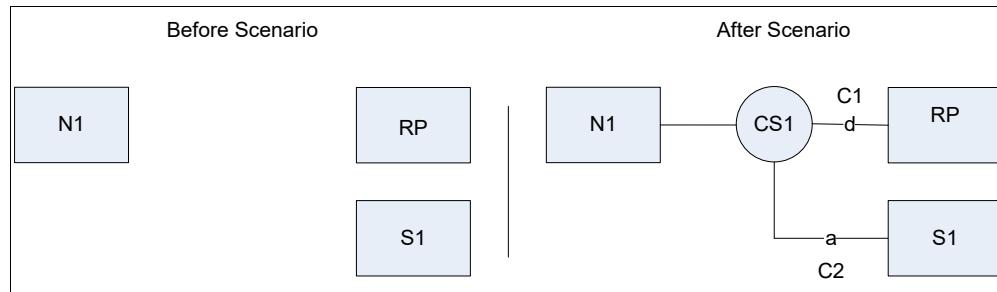


Figure 142 Call Forwarded Message Flow

Activity	Monitored RP	Monitored S1	Comments
N1 dial RP phone number. The call is automatically forwarded toward destination S1.	Route Point Call Forwarded Event (F1) Response (F2)	Call Received Event (F3) Response (F4)	The Call Received event contains the detail of the redirection performed.

F1 RoutePointCallForwardedEvent

Parameter Name	Parameter Value
requestId	48:1960
payload	Not applicable
Event	Not applicable
eventID	56a625ea-45a4-4e50-a741-6061d35f8ada
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	CTIController
subscriptionId	5f496571-8177-4257-b303-b14909ae4966
channelId	9488b2a7-abb4-4dd3-b0f4-0ea7ea4ce0e2
targetId	RoutePoint@broadsoft.com
eventData	Not applicable
callId	callhalf-78933:0
extTrackingId	303:1
routePointName	RoutePointName
routePointNumber	tel:+15006001030



Parameter Name	Parameter Value
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
redirect	Not applicable
address	tel:1011
reason	deflection
redirectTime	1271882708100

F2 EventResponse

Parameter Name	Parameter Value
requestId	48:1960
payload	Not applicable
EventResponse	Not applicable
eventID	56a625ea-45a4-4e50-a741-6061d35f8ada
statusCode	200
reason	OK

F3 CallReceivedEvent

Parameter Name	Parameter Value
requestId	48:1961
payload	Not applicable
Event	Not applicable
eventID	5cd8e3d2-b7a0-4ff2-a6b4-a8e11f89faa9
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	CTIController
subscriptionId	a5c6664e-d9ab-46e4-a24d-20f413bf2e1a
channelId	9488b2a7-abb4-4dd3-b0f4-0ea7ea4ce0e2
targetId	subscriberS1@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-78937:0
extTrackingId	303:1
personality	Terminator
state	Alerting

Parameter Name	Parameter Value
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
redirections	Not applicable
redirection	Not applicable
party	Not applicable
name	RoutePointFirstName RoutePointLastName
address	tel:+15006001030
userId	RoutePoint@broadsoft.com
callType	Group
reason	deflection
startTime	1271882708100

F4 EventResponse

Parameter Name	Parameter Value
requestId	48:1961
payload	Not applicable
EventResponse	Not applicable
eventID	5cd8e3d2-b7a0-4ff2-a6b4-a8e11f89faa9
statusCode	200
reason	OK

9.3.9 Holiday Policy Applied

This scenario illustrates a case where a call is forwarded because of the applicable Holiday schedule. In this scenario, network party N1 initiates a call toward the Route Point RP for which the Holiday schedule policy is enabled with redirection to S1. Note that the Night policy scenario is similar to the current one and as such is not shown.

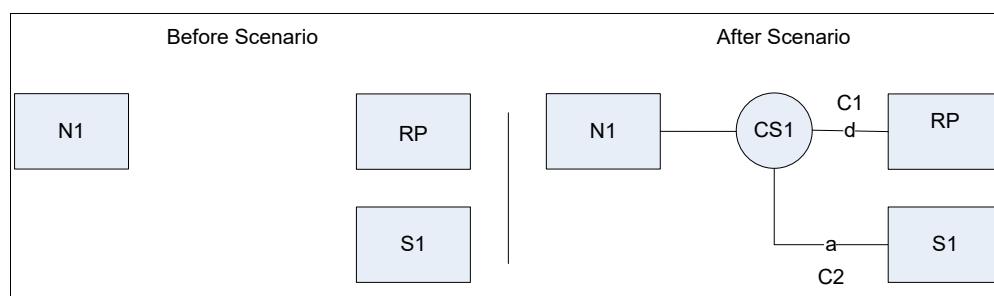


Figure 143 Holiday Policy Applied Message Flow



Activity	Monitored RP	Monitored S1	Comments
N1 dials RP phone number. The call is automatically forwarded toward destination S1 because of the holiday policy.	Route Point Holiday Policy Applied (F1) Response (F2)	Call Received Event (F3) Response (F4)	The Call Received event contains the detail of the redirection performed.

F1 RoutePointHolidayPolicyAppliedEvent

Parameter Name	Parameter Value
requestId	1:2075
payload	Not applicable
Event	Not applicable
eventID	4ad5f5c0-0ce6-4c6d-b9c3-bbe8eb883e8a
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtId
subscriptionId	c09f2dee-32ab-4408-9ea9-cb9806331191
channelId	e89e1bb5-5223-4af1-849d-0e468ee9c07c
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
callId	callhalf-79297:0
extTrackingId	324:1
routePointName	RoutePointName
routePointNumber	tel:+15006001030
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
redirect	Not applicable
address	tel:1011
reason	deflection
redirectTime	1271884332716

F2 EventResponse

Parameter Name	Parameter Value
requestId	1:2075
payload	Not applicable
EventResponse	Not applicable
eventID	4ad5f5c0-0ce6-4c6d-b9c3-bbe8eb883e8a



Parameter Name	Parameter Value
statusCode	200
reason	OK

F3 CallReceivedEvent

Parameter Name	Parameter Value
requestId	1:2076
payload	Not applicable
Event	Not applicable
eventID	b4ce3b43-f1eb-4b3f-b369-c73f614004f6
sequenceNumber	29
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	CTIController
subscriptionId	a5c6664e-d9ab-46e4-a24d-20f413bf2e1a
channelId	e89e1bb5-5223-4af1-849d-0e468ee9c07c
targetId	subscriberS1@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-79301:0
extTrackingId	324:1
personality	Terminator
state	Alerting
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
redirections	Not applicable
redirection	Not applicable
party	Not applicable
name	RoutePointFirstName RoutePointLastName
address	tel:+15006001030
userId	RoutePoint@broadsoft.com
callType	Group
reason	deflection
startTime	1271884332717

F4 EventResponse

Parameter Name	Parameter Value
requestId	1:2076
payload	Not applicable
EventResponse	Not applicable
eventId	b4ce3b43-f1eb-4b3f-b369-c73f614004f6
statusCode	200
reason	OK

9.3.10 Monitor Next Route Point

This scenario illustrates a successful Monitor Next Call. In this scenario, subscriber S1 initiates a monitor next call on route point RP. Network party N1 then initiates a call toward route point RP, which is monitored by S1.

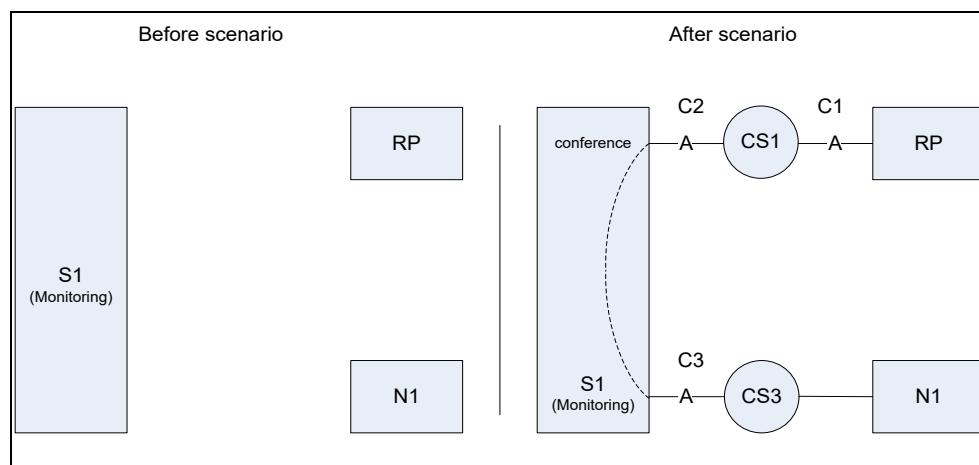


Figure 144 Monitor Next Route Point Message Flow

Activity	Monitored Subscriber S1	Monitored Route Point RP	Comments
A Monitor Next Call is invoked on behalf of subscriber S1.	Monitor Next Call Request (F1) Response (F2)		
Indication that the service has been initiated from this subscriber.	Call Originating Event (F3) Response (F4)		The “Click-to-Dial” personality indicates that the subscriber S1 is being prompted (via ringing, for example) to go off-hook.
Indication that the subscriber’s endpoint supports the SIP talk event package.	Call Updated Event (F5) Response (F6)		



Activity	Monitored Subscriber S1	Monitored Route Point RP	Comments
Subscriber S1 goes off-hook and receives silence until the next received call by the route point.	Hook Status Event (F7) Response (F8) Call Collecting Event (F9) Response (F10)		
Network party N1 dials the Route Point number. The Route Point receives the new call and monitoring is started.	Call Originated Event (F11) Response (F12)		
Indicates that call C3 is originated for the monitor.	Call Originated Event (F13) Response (F14)		
Indicates that call C2 is now Active.	Call Answered Event (F15) Response (F16)		
Indicates that call C3 personality has changed to Terminator.	Call Updated (F17) Response (F18)		
Indicates that the call C2 no longer has an endpoint associated.	Call Updated (F19) Response (F20)		
Indicates that the conference is started.	Conference Started Event (F21) Response (F22)		Subscriber S1 is muted. Network Party and the Route Point communicate without hearing S1. Subscriber S1 can hear both participants.
An incoming call is received at the Route Point and is screened against the incoming call policy. It is then added to the Route Point queue.		Route Point Call Added Event (F23) Response (F24)	
Indicates that call C3 is now Active.	Call Answered Event (F25) Response (F26)		
A Route Point Play Treatment is invoked on the newly added call.		Route Point Play Treatment Request (F27) Response (F28)	Subscriber S1 hears the treatment.
		Route Point Play Treatment Started Event (F29) Response (F30)	
The treatment is completed.		Event: Route Point Treatment Completed (F31) Response (F32)	



F1 Request

Parameter Name	Parameter Value
requestId	11
sessionId	23
MonitorNextCallRequest	N/A
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/silentmonitornext?%address%&%location%
method	POST
version	N/A
params	N/A
location	All
address	tel:1030
subscriberId	subscriberS1@broadworks.com
payload	N/A

F2 Response

Parameter Name	Parameter Value
requestId	11
sessionId	23
statusCode	201
reason	Created
payload	N/A
CallStartInfo	N/A
callId	callhalf-17175:0
externalTrackingId	578b6cfb-59fe-429b-8aae-090f9e68adf9

F3 CallOriginatingEvent

Parameter Name	Parameter Value
requestId	0:5242
payload	N/A
Event	N/A
eventID	004e6efe-2b90-4916-b339-0361160afc41
sequenceNumber	53
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	84e09bd8-0063-48a0-9f79-d6cd58a55230
channelId	a2bdcaae-296e-4c95-b70e-b8e7cf568421



Parameter Name	Parameter Value
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-17175:0
extTrackingId	578b6cfb-59fe-429b-8aae-090f9e68adf9
personality	Click-to-Dial
state	Alerting
remoteParty	N/A
address	tel:#841030
callType	Unknown
appearance	1
startTime	1382370049281

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:5242
payload	N/A
EventResponse	N/A
eventID	004e6efe-2b90-4916-b339-0361160afc41
statusCode	200
reason	OK

F5 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:5243
payload	N/A
Event	N/A
eventID	efcaca88-bf59-463a-b626-38819d1650b9
sequenceNumber	54
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	84e09bd8-0063-48a0-9f79-d6cd58a55230
channelId	a2bdcaa-e-296e-4c95-b70e-b8e7cf568421
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A



Parameter Name	Parameter Value
callId	callhalf-17175:0
extTrackingId	578b6cfb-59fe-429b-8aae-090f9e68adf9
personality	Click-to-Dial
state	Alerting
remoteParty	N/A
address	tel:#841030
callType	Unknown
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
allowAnswer	N/A
startTime	1382370049281

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:5243
payload	N/A
EventResponse	N/A
eventID	efcaca88-bf59-463a-b626-38819d1650b9
statusCode	200
reason	OK

F7 HookStatusEvent

Parameter Name	Parameter Value
requestId	0:5244
payload	N/A
Event	N/A
eventID	52cf9b7e-438b-422b-ba01-c0e8f4c32e2a
sequenceNumber	55
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	84e09bd8-0063-48a0-9f79-d6cd58a55230
channelId	a2bdcaae-296e-4c95-b70e-b8e7cf568421
targetId	subscriberS1@broadworks.com
eventData	N/A
hookStatus	Off-Hook



F8 EventResponse

Parameter Name	Parameter Value
requestId	0:5244
payload	N/A
EventResponse	N/A
eventID	52cf9b7e-438b-422b-ba01-c0e8f4c32e2a
statusCode	200
reason	OK

F9 CallCollectingEvent

Parameter Name	Parameter Value
requestId	0:5245
payload	N/A
Event	N/A
eventID	ba72256f-4988-4bff-9d36-e56132e87916
sequenceNumber	56
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	84e09bd8-0063-48a0-9f79-d6cd58a55230
channelId	a2bdcaee-296e-4c95-b70e-b8e7cf568421
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-17175:0
extTrackingId	578b6cfb-59fe-429b-8aae-090f9e68adf9
personality	Originator
state	Active
remoteParty	N/A
address	tel:#841030
callType	Unknown
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
startTime	1382370050469



F10 EventResponse

Parameter Name	Parameter Value
requestId	0:5245
payload	N/A
EventResponse	N/A
eventID	ba72256f-4988-4bff-9d36-e56132e87916
statusCode	200
reason	OK

F11 CallOriginatedEvent

Parameter Name	Parameter Value
requestId	0:5247
payload	N/A
Event	N/A
eventID	a72314d2-64b1-4118-b372-517a0686608b
sequenceNumber	57
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	84e09bd8-0063-48a0-9f79-d6cd58a55230
channelId	a2bdcaae-296e-4c95-b70e-b8e7cf568421
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-17175:0
extTrackingId	578b6cfb-59fe-429b-8aae-090f9e68adf9
personality	Originator
state	Alerting
remoteParty	N/A
address	tel:#841030
callType	Unknown
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
startTime	1382370050469



F12 EventResponse

Parameter Name	Parameter Value
requestId	0:5247
payload	N/A
EventResponse	N/A
eventID	a72314d2-64b1-4118-b372-517a0686608b
statusCode	200
reason	OK

F13 CallOriginatedEvent

Parameter Name	Parameter Value
requestId	0:5248
payload	N/A
Event	N/A
eventID	65ff18d6-9fdf-499a-bc8b-3d55fa843542
sequenceNumber	58
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	84e09bd8-0063-48a0-9f79-d6cd58a55230
channelId	a2bdcaae-296e-4c95-b70e-b8e7cf568421
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-17175:1
extTrackingId	9fb23595-b2f0-4b2d-86d4-24ed32fcfae4
networkCallId	NDIwMjEyNjkzMmY5MWY5NGVhMzEwZjNhMjMyYmUyOWY.
personality	Originator
state	Alerting
remoteParty	N/A
name	5007003333
address	tel:+15007003333
callType	Network
startTime	1382370054652



F14 EventResponse

Parameter Name	Parameter Value
requestId	0:5248
payload	N/A
EventResponse	N/A
eventID	65ff18d6-9fdf-499a-bc8b-3d55fa843542
statusCode	200
reason	OK

F15 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:5249
payload	N/A
Event	N/A
eventID	3d28824b-db97-45f3-9f22-9484d07d6d37
sequenceNumber	59
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	84e09bd8-0063-48a0-9f79-d6cd58a55230
channelId	a2bdcaae-296e-4c95-b70e-b8e7cf568421
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-17175:0
extTrackingId	578b6cfb-59fe-429b-8aae-090f9e68adf9
personality	Originator
state	Active
remoteParty	N/A
name	RoutePointNameFirstname RoutePointNameLastname
address	tel:1030
userId	routePoint@broadworks.com
userDN	tel:+15006001030;ext=1030
callType	Group
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
startTime	1382370050469



Parameter Name	Parameter Value
answerTime	1382370054652

F16 EventResponse

Parameter Name	Parameter Value
requestId	0:5249
payload	N/A
EventResponse	N/A
eventID	3d28824b-db97-45f3-9f22-9484d07d6d37
statusCode	200
reason	OK

F17 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:5250
payload	N/A
Event	N/A
eventID	d9bdc194-3a46-4f81-895a-fe6462e08140
sequenceNumber	60
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	84e09bd8-0063-48a0-9f79-d6cd58a55230
channelId	a2bdcaae-296e-4c95-b70e-b8e7cf568421
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-17175:1
extTrackingId	9fb23595-b2f0-4b2d-86d4-24ed32fcfae4
networkCallId	NDIwMjEyNjkwMmY5MWY5NGVhMzEwZjNhMjMyYmUyOWY.
personality	Terminator
state	Alerting
remoteParty	N/A
name	5007003333
address	tel:+15007003333
callType	Network
startTime	1382370054652



F18 EventResponse

Parameter Name	Parameter Value
requestId	0:5250
payload	N/A
EventResponse	N/A
eventID	d9bdc194-3a46-4f81-895a-fe6462e08140
statusCode	200
reason	OK

F19 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:5251
payload	N/A
Event	N/A
eventID	7fe0cc60-f161-4b19-9f25-4e40060a2bd9
sequenceNumber	61
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	84e09bd8-0063-48a0-9f79-d6cd58a55230
channelId	a2bdcaae-296e-4c95-b70e-b8e7cf568421
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-17175:0
extTrackingId	578b6cfb-59fe-429b-8aae-090f9e68adf9
personality	Originator
state	Active
remoteParty	N/A
name	RoutePointNameFirstname RoutePointNameLastname
address	tel:1030
userId	routePoint@broadworks.com
userDN	tel:+15006001030;ext=1030
callType	Group
startTime	1382370050469
answerTime	1382370054652



F20 EventResponse

Parameter Name	Parameter Value
requestId	0:5251
payload	N/A
EventResponse	N/A
eventID	7fe0cc60-f161-4b19-9f25-4e40060a2bd9
statusCode	200
reason	OK

F21 ConferenceStartedEvent

Parameter Name	Parameter Value
requestId	0:5252
payload	N/A
Event	N/A
eventID	02ebdbdf-28fc-49b2-b296-f554e22e0ee6
sequenceNumber	62
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	84e09bd8-0063-48a0-9f79-d6cd58a55230
channelId	a2bdcaae-296e-4c95-b70e-b8e7cf568421
targetId	subscriberS1@broadworks.com
EventData	N/A
conference	N/A
state	Active
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
muted	N/A
conferenceType	Silently Monitored
conferenceParticipantList	N/A
conferenceParticipant	N/A
callId	callhalf-17175:0
conferenceParticipant	N/A
callId	callhalf-17175:1



F22 EventResponse

Parameter Name	Parameter Value
requestId	0:5252
payload	N/A
EventResponse	N/A
eventID	02ebdbdf-28fc-49b2-b296-f554e22e0ee6
statusCode	200
reason	OK

F23 RoutePointCallAddedEvent

Parameter Name	Parameter Value
requestId	0:5253
payload	N/A
Event	N/A
eventID	1e0595fd-2bd1-43dc-82ba-6909b5e566fe
sequenceNumber	18
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	a7248761-05c8-47b1-9921-91addd930a44
channelId	a2bdcaae-296e-4c95-b70e-b8e7cf568421
targetId	routePoint@broadworks.com
EventData	N/A
queueEntry	N/A
callId	callhalf-17181:0
extTrackingId	578b6cfb-59fe-429b-8aae-090f9e68adf9
remoteParty	N/A
name	5007003333
address	tel:+15007003333
callType	Network
addTime	1382370054418
routePointName	RoutePointName
routePointNumber	tel:+15006001030

F24 EventResponse

Parameter Name	Parameter Value
requestId	0:5253
payload	N/A



Parameter Name	Parameter Value
EventResponse	N/A
eventID	1e0595fd-2bd1-43dc-82ba-6909b5e566fe
statusCode	200
reason	OK

F25 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:5254
payload	N/A
Event	N/A
eventID	bf050286-ba39-42b2-9e5f-8ac50677fcc7
sequenceNumber	63
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	84e09bd8-0063-48a0-9f79-d6cd58a55230
channelId	a2bdcaae-296e-4c95-b70e-b8e7cf568421
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-17175:1
extTrackingId	9fb23595-b2f0-4b2d-86d4-24ed32fcfae4
networkCallId	NDIwMjEyNjkwMmY5MWY5NGVhMzEwZjNhMjMyYmUyOWY.
personality	Terminator
state	Active
remoteParty	N/A
name	5007003333
address	tel:+15007003333
callType	Network
startTime	1382370054652
answerTime	1382370054652

F26 EventResponse

Parameter Name	Parameter Value
requestId	0:5254
payload	N/A
EventResponse	N/A



Parameter Name	Parameter Value
eventID	bf050286-ba39-42b2-9e5f-8ac50677fcc7
statusCode	200
reason	OK

F27 Request

Parameter Name	Parameter Value
requestId	12
sessionId	23
credentials	YXNvdXRoQG10bGFzZGV2ODcubmV0Om10bGxhYg==
RoutePointPlayTreatmentRequest	N/A
uri	/com.broadsoft.xsi-actions/v2.0/routePoint/%subscriberId%/%callId%/playTreatment
method	PUT
version	N/A
params	N/A
callId	callhalf-17181:0
subscriberId	routePoint@broadworks.com
payload	N/A
RoutePointPlayTreatment	N/A
audioUrlList	N/A
uri	http://192.168.8.52//media/en/QueueDefaultEntranceNoEscape.wav
videoUrlList	N/A
uri	http://192.168.8.125/media/en/QueueDefaultEntranceNoEscape.mov
numberOfPlay	1

F28 Response

Parameter Name	Parameter Value
requestId	12
sessionId	23
statusCode	200



Parameter Name	Parameter Value
reason	OK
payload	N/A
RoutePointPlayInfo	N/A
playCollectId	0

F29 RoutePointTreatmentStartedEvent

Parameter Name	Parameter Value
requestId	0:5265
payload	N/A
Event	N/A
eventID	12adc5e1-cd1c-455f-b8d4-02935ba24fe1
sequenceNumber	19
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	a7248761-05c8-47b1-9921-91addd930a44
channelId	a2bdcaae-296e-4c95-b70e-b8e7cf568421
targetId	routePoint@broadworks.com
eventData	N/A
queueEntry	N/A
callId	callhalf-17181:0
extTrackingId	578b6cfb-59fe-429b-8aae-090f9e68adf9
remoteParty	N/A
name	5007003333
address	tel:+15007003333
callType	Network
addTime	1382370054418
routePointName	RoutePointName
routePointNumber	tel:+15006001030
playCollectInfo	N/A
playCollectId	0
playCollectType	Treatment

F30 EventResponse

Parameter Name	Parameter Value
requestId	0:5265
payload	N/A
EventResponse	N/A



Parameter Name	Parameter Value
eventID	12adc5e1-cd1c-455f-b8d4-02935ba24fe1
statusCode	200
reason	OK

F31 RoutePointTreatmentCompletedEvent

Parameter Name	Parameter Value
requestId	0:5267
payload	N/A
Event	N/A
eventID	d85fc0b1-31e5-4ca7-bff7-2cf74ee1af86
sequenceNumber	20
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	a7248761-05c8-47b1-9921-91addd930a44
channelId	a2bdcaae-296e-4c95-b70e-b8e7cf568421
targetId	routePoint@broadworks.com
eventData	N/A
queueEntry	N/A
callId	callhalf-17181:0
extTrackingId	578b6cfb-59fe-429b-8aae-090f9e68adf9
remoteParty	N/A
name	5007003333
address	tel:+15007003333
callType	Network
addTime	1382370054418
routePointName	RoutePointName
routePointNumber	tel:+15006001030
playCollectResult	N/A
playCollectId	0
completionReason	EOF

F32 EventResponse

Parameter Name	Parameter Value
requestId	0:5267
payload	N/A
EventResponse	N/A
eventID	d85fc0b1-31e5-4ca7-bff7-2cf74ee1af86

Parameter Name	Parameter Value
statusCode	200
reason	OK

9.3.11 Outgoing Call Distributed to Agent

This scenario illustrates the case where a remote application performs an outgoing dial to a Network Party. The call is added to a Route Point queue. Once answered, the call is distributed to agent A1, who answers the call.

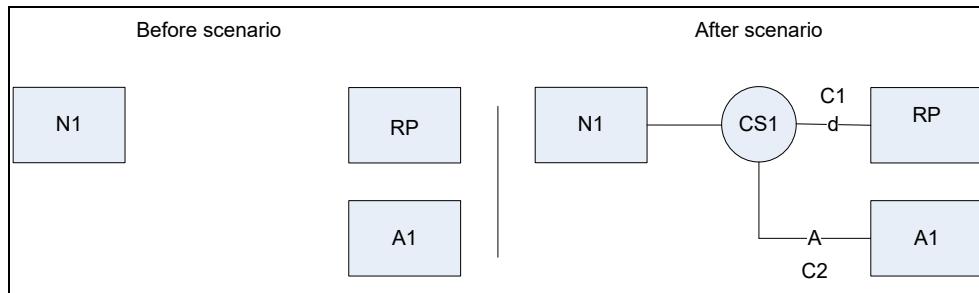


Figure 145 Outgoing Call Distributed to Agent

Activity	Monitored Route Point	Monitored Agent A1	Comments
An outgoing dial is performed to N1 phoneNumber.	Request: Route Point Outgoing Dial Request (F1) Response (F3)		The Request response contains the externalTrackingId and callId of the created Call C1.
Indication that an outgoing dial attempt is being made.	Event: Route Point Call Added Event (F2) Response (F4)		
Indication that the remote party information is available.	Route Point Call Updated Event (F5) Response (F6)		
Indication that the call is placed over the network.	Event: Route Point Call Originated (F7) Response (F8)		
The Network Party answers the call.	Event: Route Point Outgoing Call Answered Event (F9) Response (F10)		The call is answered and silence is being played to the Network Party N1.
A Route Point Distribute Call is performed on the newly added call to Agent A1.	Request: Route Point Distribute Call Request (F11) Response (F14)		No ringback is played to N1.
The call is offered to the agent. Agent A1 phone begins to ring.	Event: Route Point Call Offered to Agent Event (F12) Response (F13)	Event: Call Received Event (F15) Response (F16)	
The agent A1 answers the call.	Event: Route Point Call Answered by Agent Event (F21) Response (F22)	Event: Hook Status Event (F17) Response (F18) Event: Call Answered Event (F19) Response (F20)	The call is taken out of the Route Point queue.

F1 Request

Parameter Name	Parameter Value
requestId	13
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
RoutePointOutgoing DialRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/routepoint/%subscriberId%/calls/new
method	POST
version	Not applicable
params	Not applicable
subscriberId	RoutePoint@broadsoft.com
payload	Not applicable
RoutePointOutgoing Dial	Not applicable
address	7003333
routePointName	RoutePointName

F2 RoutePointCallAddedEvent

Parameter Name	Parameter Value
requestId	0:6115
payload	Not applicable
Event	Not applicable
eventID	c562639e-0b10-4154-acce-a70b2537ab34
sequenceNumber	17
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	99e6fd92-02b3-45e9-9ef5-5a963dba68a7
channelId	d0b58538-045a-4afc-8fe3-51ad1d418cd3
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-46773:0
extTrackingId	627:1
remoteParty	Not applicable
callType	Unknown
addTime	1271682060161



Parameter Name	Parameter Value
routePointName	RoutePointName
routePointNumber	tel:+15006001030
outgoingCall	Not applicable

F3 EventResponse

Parameter Name	Parameter Value
requestId	0:6115
payload	Not applicable
EventResponse	Not applicable
eventID	c562639e-0b10-4154-acce-a70b2537ab34
statusCode	200
reason	OK

F4 Response

Parameter Name	Parameter Value
requestId	13
sessionId	514
statusCode	201
reason	Not applicable
payload	Not applicable
CallStartInfo	Not applicable
callId	callhalf-46773:0
externalTrackingId	627:1

F5 RoutePointCallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:6117
payload	Not applicable
Event	Not applicable
eventID	daec1f72-6c52-46e2-85d5-feedf0f54508
sequenceNumber	18
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtId
subscriptionId	99e6fd92-02b3-45e9-9ef5-5a963dba68a7
channelId	d0b58538-045a-4afc-8fe3-51ad1d418cd3
targetId	RoutePoint@broadsoft.com
EventData	Not applicable



Parameter Name	Parameter Value
queueEntry	Not applicable
callId	callhalf-46773:0
extTrackingId	627:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271682060161
routePointName	RoutePointName
routePointNumber	tel:+15006001030
outgoingCall	Not applicable
reason	Unknown

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:6117
payload	Not applicable
EventResponse	Not applicable
eventID	daec1f72-6c52-46e2-85d5-feedf0f54508
statusCode	200
reason	OK

F7 RoutePointOutgoingCallOriginatedEvent

Parameter Name	Parameter Value
requestId	0:6118
payload	Not applicable
Event	Not applicable
eventID	02b4d14d-5355-40fc-8e12-52fff7b9c0cb
sequenceNumber	19
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	99e6fd92-02b3-45e9-9ef5-5a963dba68a7
channelId	d0b58538-045a-4afc-8fe3-51ad1d418cd3
targetId	RoutePoint@broadsoft.com
eventData	Not applicable
queueEntry	Not applicable



Parameter Name	Parameter Value
callId	callhalf-46773:0
extTrackingId	627:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271682060161
routePointName	RoutePointName
routePointNumber	tel:+15006001030
outgoingCall	Not applicable

F8 EventResponse

Parameter Name	Parameter Value
requestId	0:6118
payload	Not applicable
EventResponse	Not applicable
eventID	02b4d14d-5355-40fc-8e12-52fff7b9c0cb
statusCode	200
reason	OK

F9 RoutePointOutgoingCallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:6119
payload	Not applicable
Event	Not applicable
eventID	00591533-1e98-412a-ae58-571ee4d8ad7f
sequenceNumber	20
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	99e6fd92-02b3-45e9-9ef5-5a963dba68a7
channelId	d0b58538-045a-4afc-8fe3-51ad1d418cd3
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-46773:0
extTrackingId	627:1



Parameter Name	Parameter Value
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271682060161
routePointName	RoutePointName
routePointNumber	tel:+15006001030
outgoingCall	Not applicable
outgoingCallAnswerTime	1271682062229

F10 EventResponse

Parameter Name	Parameter Value
requestId	0:6119
payload	Not applicable
EventResponse	Not applicable
eventID	00591533-1e98-412a-ae58-571ee4d8ad7f
statusCode	200
reason	OK

F11 Request

Parameter Name	Parameter Value
requestId	14
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
RoutePointDistribute CallRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/routePoint/%subscriberId%/calls/%callId%/distribute
method	PUT
version	Not applicable
params	Not applicable
callId	callhalf-46773:0
subscriberId	RoutePoint@broadsoft.com
payload	Not applicable
RoutePointDistribute	Not applicable
agentId	agentA1@broadsoft.com
callingLineIdNumber	tel:15006001025



Parameter Name	Parameter Value
callingLineIdName	RoutePointName
timeInQueue	1
numberOfCallsInQueue	1
longestWaitTime	1

F12 RoutePointCallOfferedToAgentEvent

Parameter Name	Parameter Value
requestId	0:6120
payload	Not applicable
Event	Not applicable
eventID	78d9b594-594f-4e01-a153-5a11742c53fa
sequenceNumber	21
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	99e6fd92-02b3-45e9-9ef5-5a963dba68a7
channelId	d0b58538-045a-4afc-8fe3-51ad1d418cd3
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-46773:0
extTrackingId	627:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271682060161
routePointName	RoutePointName
routePointNumber	tel:+15006001030
outgoingCall	Not applicable
outgoingCallAnswerTime	1271682062229

F13 EventResponse

Parameter Name	Parameter Value
requestId	0:6120
payload	Not applicable
EventResponse	Not applicable



Parameter Name	Parameter Value
eventID	78d9b594-594f-4e01-a153-5a11742c53fa
statusCode	200
reason	OK

F14 Response

Parameter Name	Parameter Value
requestId	14
sessionId	515
statusCode	200
reason	Not applicable
payload	Not applicable

F15 CallReceivedEvent

Parameter Name	Parameter Value
requestId	0:6121
payload	Not applicable
Event	Not applicable
eventID	70c0b441-2ca1-400b-be8c-623f59e49d41
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	e824a075-141c-4394-a99a-936ec46254ae
channelId	d0b58538-045a-4afc-8fe3-51ad1d418cd3
targetId	agentA1@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-46827:0
extTrackingId	627:1
personality	Terminator
state	Alerting
remoteParty	Not applicable
name	RoutePointFirstName RoutePointLastName - NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
startTime	1271682099952
acdCallInfo	Not applicable

Parameter Name	Parameter Value
acdUserId	RoutePoint@broadsoft.com
acdName	RoutePointName
acdNumber	tel:+15006001030
numCallsInQueue	0
waitTime	1
longestWaitTime	1
callingPartyInfo	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network

F16 EventResponse

Parameter Name	Parameter Value
requestId	0:6121
payload	Not applicable
EventResponse	Not applicable
eventID	70c0b441-2ca1-400b-be8c-623f59e49d41
statusCode	200
reason	OK

F17 HookStatusEvent

Parameter Name	Parameter Value
RequestId	0:6122
Payload	Not applicable
Event	Not applicable
EventID	3f5bca6d-78ef-4350-87f0-0c5203091cdd
sequenceNumber	3
UserId	admin@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	ceaec03b-a6b7-4681-b4de-287d34cc2b89
channelId	d0b58538-045a-4afc-8fe3-51ad1d418cd3
Targeted	agentA1@broadsoft.com
EventData	Not applicable
hookStatus	Off-Hook



F18 EventResponse

Parameter Name	Parameter Value
requestId	0:6122
Payload	Not applicable
EventResponse	Not applicable
eventide	3f5bca6d-78ef-4350-87f0-0c5203091cdd
statusCode	200
reason	OK

F19 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:6123
payload	Not applicable
Event	Not applicable
eventId	3f5bca6d-78ef-4350-87f0-0c5203091cdc
sequenceNumber	4
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	e824a075-141c-4394-a99a-936ec46254ae
channelId	d0b58538-045a-4afc-8fe3-51ad1d418cd3
targetId	agentA1@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-46827:0
extTrackingId	627:1
personality	Terminator
state	Active
remoteParty	Not applicable
name	RoutePointFirstName RoutePointLastName - NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
endpoint	Not applicable
addressOfRecord	5006001020@broadsoft.com
appearance	1
startTime	1271682099952
answerTime	1271682102364

Parameter Name	Parameter Value
acdCallInfo	Not applicable
acdUserId	RoutePoint@broadsoft.com
acdName	RoutePointName
acdNumber	tel:+15006001030
numCallsInQueue	0
waitTime	1
longestWaitTime	1
callingPartyInfo	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network

F20 EventResponse

Parameter Name	Parameter Value
requestId	0:6124
payload	Not applicable
EventResponse	Not applicable
eventID	3f5bca6d-78ef-4350-87f0-0c5203091cdc
statusCode	200
reason	OK

F21 RoutePointCallAnsweredByAgentEvent

Parameter Name	Parameter Value
requestId	0:6123
payload	Not applicable
Event	Not applicable
eventID	222ffe0b-47d9-42bc-820b-16792132b627
sequenceNumber	22
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	99e6fd92-02b3-45e9-9ef5-5a963dba68a7
channelId	d0b58538-045a-4afc-8fe3-51ad1d418cd3
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-46773:0



Parameter Name	Parameter Value
extTrackingId	627:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271682060161
removeTime	1271682102397
routePointName	RoutePointName
routePointNumber	tel:+15006001030
outgoingCall	Not applicable
outgoingCallAnswer Time	1271682062229
answeringUserId	agentA1@broadsoft.com
answeringCallId	callhalf-46827:0

F22 EventResponse

Parameter Name	Parameter Value
requestId	0:6123
payload	Not applicable
EventResponse	Not applicable
eventID	222ffe0b-47d9-42bc-820b-16792132b627
statusCode	200
reason	OK

9.3.12 Route Point Failure and Recovery

This scenario illustrates a case where an entire Route Point is put in the *Failed* state and then back to the *Normal* state.

In this scenario, the route point configuration for failed destination is set as the ACD. Network party N1 and N2 are initially connected to the Route Point RP via call C1 and C2. Call C1 was first added to RP queue followed by call C2. This order is important as calls are moved from RP to the ACD in a "First In First Out" manner. So call C1 is first moved to the ACD, and then call C2 is moved to the ACD. Once the route point is back to the *Normal* state, it can start queuing calls again.

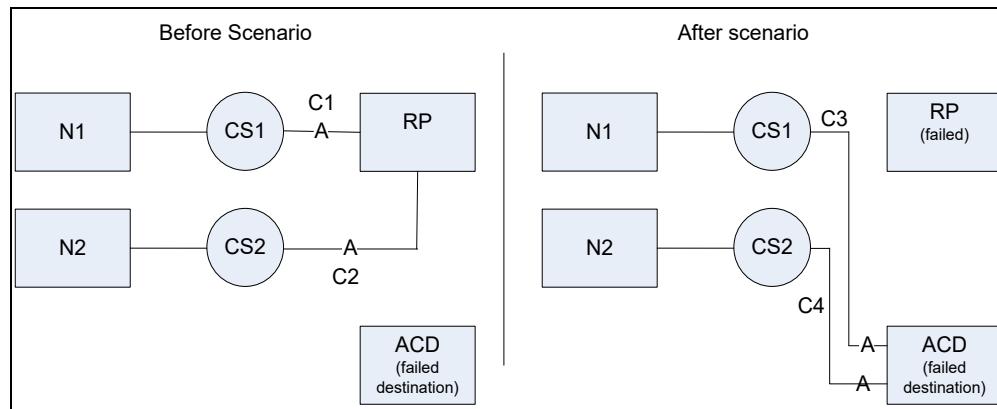


Figure 146 Route Point Failure and Recovery Message Flow

Activity	Monitored RP	Monitored ACD	Comments
A Route Point Fail is performed.	Fail Route Point Request (F1) Response (F2)		
Indicates that the Route Point has transited to the <i>Failed</i> state.	Route Point Failed Event (F3) Response (F4)		Calls will be sent to the failed destination in a "first in first out" (FIFO) manner. New incoming calls sent to the Route Point are now forwarded to the ACD.
Indicates that call C1 is removed from the queue.	Route Point Call Failed Event (F5) Response (F6)		The first call is removed from the RP queue.
Indicates that call C3 is added to the ACD queue.		ACD Call added Event (F7) Response (F8)	
Indicates that call C2 is removed from the queue.	Route Point Call Failed Event (F9) Response (F10)		The second call is removed from the RP queue.
Indicates that call C4 is added to the ACD queue.		ACD Call added Event (F11) Response (F12)	
A Route Point Recover is performed.	Route Point Recover Request (F13) Response (F14)		



Activity	Monitored RP	Monitored ACD	Comments
Indicates that the Route Point has transited to the <i>Normal</i> state.	Event: Route Point Recovered (F15) Response (F16)		New incoming calls sent to the Route Point can now be handled by the Route Point.

F1 Request

Parameter Name	Parameter Value
requestId	21
sessionId	MySessionId
credentials	YWRtaW5DVElAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
ModifyRoutePoint StateRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/routepoint/%subscriberId%/profile/state
method	PUT
version	Not applicable
params	Not applicable
subscriberId	RoutePoint@broadsoft.com
payload	Not applicable
RoutePointState	Failed

F2 Response

Parameter Name	Parameter Value
requestId	21
sessionId	865
statusCode	200
reason	OK
payload	Not applicable

F3 RoutePointFailedEvent

Parameter Name	Parameter Value
requestId	0:7014
payload	Not applicable
Event	Not applicable
eventID	e283fc20-a270-4941-8661-9db98c0304f4
sequenceNumber	22
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplication Id	CTIController
subscriptionId	270a9775-87a0-4a30-8455-cb75e5c62ec2

Parameter Name	Parameter Value
channelId	a405e60d-087d-4fc4-82ac-4ae260b81f8d
targetId	RoutePoint@broadsoft.com
eventData	Not applicable
stateChange Reason	ExternalRequest

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:7014
payload	Not applicable
EventResponse	Not applicable
eventID	e283fc20-a270-4941-8661-9db98c0304f4
statusCode	200
reason	OK

F5 RoutePointCallFailedEvent

Parameter Name	Parameter Value
requestId	0:7015
payload	Not applicable
Event	Not applicable
eventID	cfc68120-db6a-4d91-bbc3-76a8edd969de
sequenceNumber	23
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplication Id	CTIController
subscriptionId	270a9775-87a0-4a30-8455-cb75e5c62ec2
channelId	a405e60d-087d-4fc4-82ac-4ae260b81f8d
targetId	RoutePoint@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-52781:0
extTrackingId	58:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271709741395
removeTime	1271709778884



Parameter Name	Parameter Value
routePointName	RoutePointName
routePointNumber	tel:+15006001030
failureReason	RoutePointFailure

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:7015
payload	Not applicable
EventResponse	Not applicable
eventID	cfc68120-db6a-4d91-bbc3-76a8edd969de
statusCode	200
reason	OK

F7 ACDCallAddedEvent

Parameter Name	Parameter Value
requestId	0:7016
payload	Not applicable
Event	Not applicable
eventID	0ec3eca7-aacf-4afb-a641-754d29a15627
sequenceNumber	10
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	CTIController
subscriptionId	a4dfb8e0-18d1-4389-9f92-da4126894eed
channelId	a405e60d-087d-4fc4-82ac-4ae260b81f8d
targetId	ACD_CTI1@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-52797:0
extTrackingId	58:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271709778917
acdName	ACD_CTI1_Name
acdNumber	tel:+15006001025

Parameter Name	Parameter Value
acdPriority	0-Highest
addTimeInPriority Bucket	1271709778917
position	1

F8 EventResponse

Parameter Name	Parameter Value
requestId	0:7016
payload	Not applicable
EventResponse	Not applicable
eventID	0ec3eca7-aacf-4afb-a641-754d29a15627
statusCode	200
reason	OK

F9 RoutePointCallFailedEvent

Parameter Name	Parameter Value
requestId	0:7017
payload	Not applicable
Event	Not applicable
eventID	3de2f2b9-8060-410b-b03f-8fd2290f8951
sequenceNumber	24
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	CTIController
subscriptionId	270a9775-87a0-4a30-8455-cb75e5c62ec2
channelId	a405e60d-087d-4fc4-82ac-4ae260b81f8d
targetId	RoutePoint@broadsoft.com
eventData	Not applicable
queueEntry	Not applicable
callId	callhalf-52793:0
extTrackingId	60:1
remoteParty	Not applicable
name	NetworkN2FirstName NetworkN2LastName
address	tel:+15007003334
callType	Network
addTime	1271709765138
removeTime	1271709778890
routePointName	RoutePointName



Parameter Name	Parameter Value
routePointNumber	tel:+15006001030
failureReason	RoutePointFailure

F10 EventResponse

Parameter Name	Parameter Value
requestId	0:7017
payload	Not applicable
EventResponse	Not applicable
eventID	3de2f2b9-8060-410b-b03f-8fd2290f8951
statusCode	200
reason	OK

F11 ACDCallAddedEvent

Parameter Name	Parameter Value
requestId	0:7018
payload	Not applicable
Event	Not applicable
eventID	630cb90e-20d7-4b9d-8e24-9aa92582af9f
sequenceNumber	11
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	CTIController
subscriptionId	a4dfb8e0-18d1-4389-9f92-da4126894eed
channelId	a405e60d-087d-4fc4-82ac-4ae260b81f8d
targetId	ACD_CTI1@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-52801:0
extTrackingId	60:1
remoteParty	Not applicable
name	NetworkN2FirstName NetworkN2LastName
address	tel:+15007003334
callType	Network
addTime	1271709778958
acdName	ACD_CTI1_Name
acdNumber	tel:+15006001025
acdPriority	0-Highest



Parameter Name	Parameter Value
addTimeInPriority Bucket	1271709778958
position	2

F12 EventResponse

Parameter Name	Parameter Value
requestId	0:7018
payload	Not applicable
EventResponse	Not applicable
eventID	630cb90e-20d7-4b9d-8e24-9aa92582af9f
statusCode	200
reason	OK

F13 Request

Parameter Name	Parameter Value
requestId	22
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
ModifyRoutePoint StateRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/routepoint/%subscriberId%/profile/state
method	PUT
version	Not applicable
params	Not applicable
subscriberId	RoutePoint@broadsoft.com
payload	Not applicable
RoutePointState	Normal

F14 Response

Parameter Name	Parameter Value
requestId	22
sessionId	870
statusCode	200
reason	OK
payload	Not applicable



F15 RoutePointRecoveredEvent

Parameter Name	Parameter Value
requestId	0:7037
payload	Not applicable
Event	Not applicable
eventID	4352ec41-d8da-44a1-8a95-621062314618
sequenceNumber	25
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	CTIController
subscriptionId	270a9775-87a0-4a30-8455-cb75e5c62ec2
channelId	a405e60d-087d-4fc4-82ac-4ae260b81f8d
targetId	RoutePoint@broadsoft.com
EventData	Not applicable

F16 EventResponse

Parameter Name	Parameter Value
requestId	0:7037
payload	Not applicable
EventResponse	Not applicable
eventID	4352ec41-d8da-44a1-8a95-621062314618
statusCode	200
reason	OK

9.4 ACD Message Flows (AS only)

Throughout this section, the ACDQueueEntry has been changed to ACDQEntry for improved readability. The following subsections illustrate the following scenario:

- Call received with treatment and then distributed
- Bounced and Abandoned Call
- Call Overflow
- Call Forwarded
- Holiday Policy Applied
- Call Stranded
- Call Transfer
- Call Updated
- Call Promoted, Reordered and finally escaped

9.4.1 Call Received with Treatment then Distributed

This scenario illustrates the case where a network party call is added to an ACD queue and then a treatment is played to the caller. The call is then distributed to agent A1, which answers the call. Finally, during the communication, agent A1 enters a disposition code.

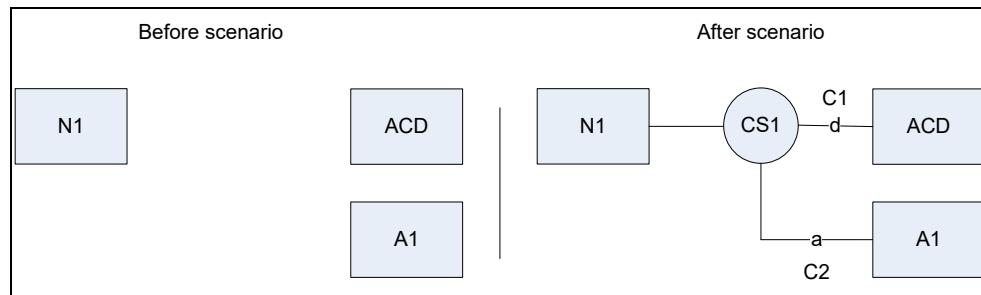


Figure 147 Call Received with Treatment and Then Distributed

Activity	Monitored ACD	Monitored Agent A1	Comments
An incoming call is received at the ACD and is screened against the incoming call policy. It is then added to the ACD queue where a treatment (entrance message) is played to the caller.	ACD Call Added Event (F1) Response (F2)		The event specifies the position of the call in the queue along with its priority.
The call is offered to an available agent. Agent A1 phone starts to ring.	ACD Call Offered to Agent Event (F5) Response (F6)	Call Received (F3) Response (F4)	Ringing is played to the caller while the call is being offered as per ACD configuration.
The agent A1 answers the call and hears the whisper message.	ACD Whisper Started Event (F9) Response (F10)	Hook Status (F7) Response (F8)	
The whisper message ends.	ACD Call Answered by Agent Event (F11) Response (F12)	Call Answered Event(F13) Response (F14)	The call is taken out of the ACD queue. The communication is established between the agent and the caller.
The agent A1 enters a disposition code to tag the call.		Enter Disposition Code Request (F15) Response (F16)	
Indication that the agent has entered a disposition code.		Agent Disposition Code Entered Event (F17) Response (F18)	

F1 ACDCallAddedEvent

Parameter Name	Parameter Value
requestId	0:6561
payload	N/A
Event	N/A
eventID	948eb411-a013-46db-ae67-d33074925de1
sequenceNumber	26



Parameter Name	Parameter Value
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	10f30f5a-6207-4399-addb-822f43cbdd0f
channelId	ab0fcfe8-b74e-463d-866e-7bb84f3156b1
targetId	ACD_CTI1@broadworks.com
eventData	N/A
queueEntry	N/A
callId	callhalf-24851:0
extTrackingId	bd01394d-eed0-4c07-acea-4e7919a6a006
remoteParty	N/A
name	5007003333
address	tel:+15007003333
callType	Network
addTime	1382474876210
acdName	ACDName
acdNumber	tel:+15006001025
acdPriority	0-Highest
addTimeInPriorityBucket	1382474876210
position	1

F2 EventResponse

Parameter Name	Parameter Value
requestId	0:6561
payload	N/A
EventResponse	N/A
eventId	948eb411-a013-46db-ae67-d33074925de1
statusCode	200
reason	OK

F3 CallReceivedEvent

Parameter Name	Parameter Value
requestId	0:6562
payload	N/A
Event	N/A
eventId	8d6ce144-e47e-470f-b888-9655153a305d
sequenceNumber	35



Parameter Name	Parameter Value
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	0ca8d6e9-7151-4323-a0c2-b22c94ec277e
channelId	ab0fcfe8-b74e-463d-866e-7bb84f3156b1
targetId	agentA1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-24855:0
extTrackingId	bd01394d-eed0-4c07-acea-4e7919a6a006
personality	Terminator
state	Alerting
remoteParty	N/A
name	ACDNameFirstname ACDNameLastname – 5007003333
address	tel:+15007003333
callType	Network
redirections	N/A
redirection	N/A
party	N/A
name	ACDNameFirstname ACDNameLastname
address	tel:+15006001025
userId	ACD_CTI1@broadworks.com
callType	Group
reason	call-center
startTime	1382474876210
acdCallInfo	N/A
acdUserId	ACD_CTI1@broadworks.com
acdName	ACDName
acdNumber	tel:+15006001025
numCallsInQueue	0
waitTime	0
callingPartyInfo	N/A
name	5007003333
address	tel:+15007003333
callType	Network



F4 EventResponse

Parameter Name	Parameter Value
requestId	0:6562
payload	N/A
EventResponse	N/A
eventID	8d6ce144-e47e-470f-b888-9655153a305d
statusCode	200
reason	OK

F5 ACDCallOfferedToAgentEvent

Parameter Name	Parameter Value
requestId	0:6563
payload	N/A
Event	N/A
eventID	ffe3e51e-9f18-40a0-a3c7-00c56775b2f4
sequenceNumber	27
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	10f30f5a-6207-4399-addb-822f43cbdd0f
channelId	ab0fcfe8-b74e-463d-866e-7bb84f3156b1
targetId	ACD_CTI1@broadworks.com
EventData	N/A
queueEntry	N/A
callId	callhalf-24851:0
extTrackingId	bd01394d-eed0-4c07-acea-4e7919a6a006
remoteParty	N/A
name	5007003333
address	tel:+15007003333
callType	Network
addTime	1382474876210
acdName	ACDName
acdNumber	tel:+15006001025
acdPriority	0-Highest
addTimeInPriorityBucket	1382474876210



F6 EventResponse

Parameter Name	Parameter Value
requestId	0:6563
payload	N/A
EventResponse	N/A
eventID	ffe3e51e-9f18-40a0-a3c7-00c56775b2f4
statusCode	200
reason	OK

F7 HookStatusEvent

Parameter Name	Parameter Value
requestId	0:6564
payload	N/A
Event	N/A
eventID	9d647113-7129-4d05-bc6d-5bc80184b0b1
sequenceNumber	36
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	0ca8d6e9-7151-4323-a0c2-b22c94ec277e
channelId	ab0fcfe8-b74e-463d-866e-7bb84f3156b1
targetId	agentA1@broadworks.com
eventData	N/A
hookStatus	Off-Hook

F8 EventResponse

Parameter Name	Parameter Value
requestId	0:6564
payload	N/A
EventResponse	N/A
eventID	9d647113-7129-4d05-bc6d-5bc80184b0b1
statusCode	200
reason	OK

F9 ACDWhisperStartedEvent

Parameter Name	Parameter Value
requestId	0:6565
payload	N/A
Event	N/A



Parameter Name	Parameter Value
eventID	352f6151-d846-485f-b06c-96f39e9bbe3c
sequenceNumber	28
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	10f30f5a-6207-4399-addb-822f43cbdd0f
channelId	ab0fcfe8-b74e-463d-866e-7bb84f3156b1
targetId	ACD_CTI1@broadworks.com
EventData	N/A
queueEntry	N/A
callId	callhalf-24851:0
extTrackingId	bd01394d-eed0-4c07-acea-4e7919a6a006
remoteParty	N/A
name	5007003333
address	tel:+15007003333
callType	Network
addTime	1382474876210
removeTime	1382474877904
acdName	ACDName
acdNumber	tel:+15006001025
acdPriority	0-Highest
addTimeInPriorityBucket	1382474876210
answeringUserId	agentA1@broadworks.com
answeringCallId	callhalf-24855:0

F10 EventResponse

Parameter Name	Parameter Value
requestId	0:6565
payload	N/A
EventResponse	N/A
eventID	352f6151-d846-485f-b06c-96f39e9bbe3c
statusCode	200
reason	OK



F11 ACDCallAnsweredByAgentEvent

Parameter Name	Parameter Value
requestId	0:6566
payload	N/A
Event	N/A
eventID	25cbb58a-46a3-4cc1-b7e8-102fd68eddf
sequenceNumber	29
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	10f30f5a-6207-4399-addb-822f43cbdd0f
channelId	ab0fcfe8-b74e-463d-866e-7bb84f3156b1
targetId	ACD_CTI1@broadworks.com
EventData	N/A
queueEntry	N/A
callId	callhalf-24851:0
extTrackingId	bd01394d-eed0-4c07-acea-4e7919a6a006
remoteParty	N/A
name	5007003333
address	tel:+15007003333
callType	Network
addTime	1382474876210
removeTime	1382474877904
acdName	ACDName
acdNumber	tel:+15006001025
acdPriority	0-Highest
addTimeInPriorityBucket	1382474876210
answeringUserId	agentA1@broadworks.com
answeringCallId	callhalf-24855:0

F12 EventResponse

Parameter Name	Parameter Value
requestId	0:6566
payload	N/A
EventResponse	N/A
eventID	25cbb58a-46a3-4cc1-b7e8-102fd68eddf
statusCode	200



Parameter Name	Parameter Value
reason	OK

F13 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:6567
payload	N/A
Event	N/A
eventID	a92e93a9-33a9-4095-9f2b-b4bf1e92eb03
sequenceNumber	37
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	0ca8d6e9-7151-4323-a0c2-b22c94ec277e
channelId	ab0fcfe8-b74e-463d-866e-7bb84f3156b1
targetId	agentA1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-24855:0
extTrackingId	bd01394d-eed0-4c07-acea-4e7919a6a006
personality	Terminator
state	Active
remoteParty	N/A
name	5007003333
address	tel:+15007003333
callType	Network
endpoint	N/A
addressOfRecord	5006001020@broadworks.com
appearance	1
startTime	1382474876210
answerTime	1382474877895
acdCallInfo	N/A
acdUserId	ACD_CTI1@broadworks.com
acdName	ACDName
acdNumber	tel:+15006001025
numCallsInQueue	0
waitTime	0
callingPartyInfo	N/A



Parameter Name	Parameter Value
name	5007003333
address	tel:+15007003333
callType	Network

F14 EventResponse

Parameter Name	Parameter Value
requestId	0:6567
payload	N/A
EventResponse	N/A
eventID	a92e93a9-33a9-4095-9f2b-b4bf1e92eb03
statusCode	200
reason	OK

F15 Request

Parameter Name	Parameter Value
requestId	38
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
TagDispositionCode Request	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/%callId%/addDispositionCode/%codename%
method	PUT
version	Not applicable
params	Not applicable
callId	callhalf-24855:0
codename	Promotion
subscriberId	agentA1@broadsoft.com
payload	Not applicable

F16 Response

Parameter Name	Parameter Value
requestId	38
sessionId	912
statusCode	200
reason	OK
payload	Not applicable



F17 AgentDispositionCodeAddedEvent

Parameter Name	Parameter Value
requestId	0:6568
payload	Not applicable
Event	Not applicable
eventID	c45f0448-981b-4a00-a632-3f97c1bb1a76
sequenceNumber	38
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	CTIController
subscriptionId	0ca8d6e9-7151-4323-a0c2-b22c94ec277e
channelId	ab0fcfe8-b74e-463d-866e-7bb84f3156b1
targetId	agentA1@broadsoft.com
eventData	Not applicable
dispositionCode	Promotion
callId	callhalf-24855:0

F18 EventResponse

Parameter Name	Parameter Value
requestId	0:6568
payload	Not applicable
EventResponse	Not applicable
eventID	c45f0448-981b-4a00-a632-3f97c1bb1a76
statusCode	200
reason	OK

9.4.2 Bounced and Abandoned Call

This scenario illustrates a case where a call offered to an agent is bounced because it is not answered within the amount of time specified in the bounced call policy. The call is eventually dropped by the caller.

In this scenario, network party N1 is initially connected to the ACD via call C1. Call C1 is maintained in the ACD queue.

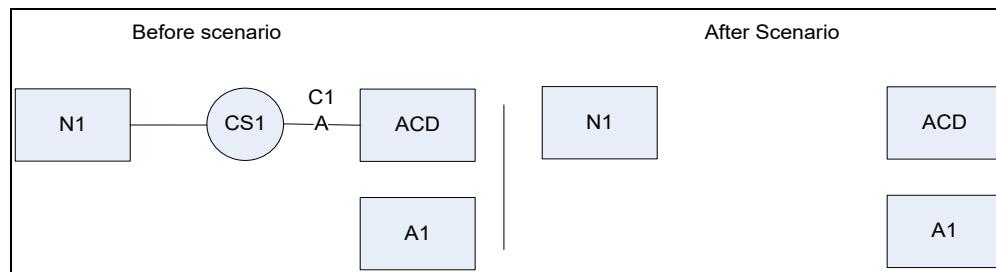


Figure 148 Bounced and Abandoned Call Message Flow

Activity	Monitored ACD	Monitored Agent A1	Comments
Agent A1 becomes available using the web portal.		Agent State Event (F1) Response (F2)	
The call is offered to the agent A1, which is now available. Agent A1 phone begins to ring.	ACD Call Offered to Agent Event (F3) Response (F4)	Call Received Event (F5) Response (F6)	Ringing is played to the caller while the call is being offered as per ACD configuration.
Agent A1 decides to set its state to the <i>Unavailable</i> state using the web portal.		Agent State Event (F7) Response (F8)	
The call is bounced and maintained in the queue.	ACD Call Bounced Event (F9) Response (F10)	Call Released Event (F11) Response (F12)	The agent did not answer the call within the amount of time specified in the bounced call policy. As per the policy, the call is simply maintained in the queue.
Caller N1 hangs up the phone.	ACD Call Abandoned Event (F13) Response (F14)		

F1 AgentStateEvent

Parameter Name	Parameter Value
requestId	25:7157
payload	Not applicable
Event	Not applicable
eventID	f9aa9e7a-3413-40c9-9331-f75b654f9dbf
sequenceNumber	6
userId	adminCTI@broadsoft.com@broadsoft.com



Parameter Name	Parameter Value
externalApplicationId	CTIController
subscriptionId	725e0c55-265a-49b4-97e8-e3a309b8c88a
channelId	5616863d-9336-4296-921a-1fb1fce69705
targetId	agentA1@broadsoft.com
eventData	Not applicable
agentStateInfo	Not applicable
state	Available
stateTimestamp	1271712574311

F2 EventResponse

Parameter Name	Parameter Value
requestId	25:7157
payload	Not applicable
EventResponse	Not applicable
eventID	f9aa9e7a-3413-40c9-9331-f75b654f9dbf
statusCode	200
reason	OK

F3 ACDCallOfferedToAgentEvent

Parameter Name	Parameter Value
requestId	25:7158
payload	Not applicable
Event	Not applicable
eventID	5df03a45-ebe8-47b8-aca0-42bea44152e4
sequenceNumber	26
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	CTIController
subscriptionId	80208e52-c673-4041-a954-0a1d91b72802
channelId	5616863d-9336-4296-921a-1fb1fce69705
targetId	ACD_CTI1@broadsoft.com
eventData	Not applicable
queueEntry	Not applicable
callId	callhalf-53197:0
extTrackingId	94:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName

Parameter Name	Parameter Value
address	tel:+15007003333
callType	Network
addTime	1271712495635
bounced	Not applicable
acdName	ACD_CTI1_Name
acdNumber	tel:+15006001025
acdPriority	0-Highest
addTimeInPriority Bucket	1271712495635

F4 EventResponse

Parameter Name	Parameter Value
requestId	25:7158
payload	Not applicable
EventResponse	Not applicable
eventID	5df03a45-ebe8-47b8-aca0-42bea44152e4
statusCode	200
reason	OK

F5 CallReceivedEvent

Parameter Name	Parameter Value
requestId	25:7159
payload	Not applicable
Event	Not applicable
eventID	ceaf115b-6e1b-453e-a890-d461dd3c39e6
sequenceNumber	15
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	CTIController
subscriptionId	5f450305-d08d-457a-9384-3b75ef0d2b60
channelId	5616863d-9336-4296-921a-1fb1fce69705
targetId	agentA1@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-53213:0
extTrackingId	94:1
personality	Terminator
state	Alerting

Parameter Name	Parameter Value
remoteParty	Not applicable
name	ACD_CTI1_FirstName ACD_CTI1_LastName - NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
redirections	Not applicable
redirection	Not applicable
party	Not applicable
name	ACD_CTI1_FirstName ACD_CTI1_LastName
address	tel:+15006001025
userId	ACD_CTI1@broadsoft.com
callType	Group
reason	call-center
startTime	1271712574324
acdCallInfo	Not applicable
acdUserId	ACD_CTI1@broadsoft.com
acdName	ACD_CTI1_Name
acdNumber	tel:+15006001025
numCallsInQueue	0
waitTime	78
callingPartyInfo	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network

F6 EventResponse

Parameter Name	Parameter Value
requestId	25:7159
payload	Not applicable
EventResponse	Not applicable
eventID	ceaf115b-6e1b-453e-a890-d461dd3c39e6
statusCode	200
reason	OK



F7 AgentStateEvent

Parameter Name	Parameter Value
requestId	25:7160
payload	Not applicable
Event	Not applicable
eventID	66f34b5e-bcb4-4faf-8b14-6db1e383d1e7
sequenceNumber	7
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	CTIController
subscriptionId	725e0c55-265a-49b4-97e8-e3a309b8c88a
channelId	5616863d-9336-4296-921a-1fb1fce69705
targetId	agentA1@broadsoft.com
eventData	Not applicable
agentStateInfo	Not applicable
state	Unavailable
stateTimestamp	1271712584635

F8 EventResponse

Parameter Name	Parameter Value
requestId	25:7160
payload	Not applicable
EventResponse	Not applicable
eventID	66f34b5e-bcb4-4faf-8b14-6db1e383d1e7
statusCode	200
reason	OK

F9 ACDCallBouncedEvent

Parameter Name	Parameter Value
requestId	25:7161
payload	Not applicable
Event	Not applicable
eventID	6121ec09-20ef-401e-9265-dab6019dcff9
sequenceNumber	27
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	CTIController
subscriptionId	80208e52-c673-4041-a954-0a1d91b72802
channelId	5616863d-9336-4296-921a-1fb1fce69705



Parameter Name	Parameter Value
targetId	ACD_CTI1@broadsoft.com
eventData	Not applicable
queueEntry	Not applicable
callId	callhalf-53197:0
extTrackingId	94:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271712495635
bounced	Not applicable
acdName	ACD_CTI1_Name
acdNumber	tel:+15006001025
acdPriority	0-Highest
addTimeInPriority Bucket	1271712495635

F10 EventResponse

Parameter Name	Parameter Value
requestId	25:7161
payload	Not applicable
EventResponse	Not applicable
eventID	6121ec09-20ef-401e-9265-dab6019dcff9
statusCode	200
reason	OK

F11 CallReleasedEvent

Parameter Name	Parameter Value
requestId	25:7162
payload	Not applicable
Event	Not applicable
eventID	1042c249-f8be-4914-9011-fbd6cdf4f9b7
sequenceNumber	16
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	CTIController
subscriptionId	5f450305-d08d-457a-9384-3b75ef0d2b60
channelId	5616863d-9336-4296-921a-1fb1fce69705

Parameter Name	Parameter Value
targetId	agentA1@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-53213:0
extTrackingId	94:1
personality	Terminator
state	Released
releasingParty	remoteRelease
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
endpoint	Not applicable
addressOfRecord	5006001020@broadsoft.com
startTime	1271712574324
releaseTime	1271712586330
acdCallInfo	Not applicable
acdUserId	ACD_CTI1@broadsoft.com
acdName	ACD_CTI1_Name
acdNumber	tel:+15006001025
numCallsInQueue	0
waitTime	78
callingPartyInfo	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network

F12 EventResponse

Parameter Name	Parameter Value
requestId	25:7162
payload	Not applicable
EventResponse	Not applicable
eventID	1042c249-f8be-4914-9011-fbd6cdf4f9b7
statusCode	200
reason	OK



F13 ACDCallAbandonedEvent

Parameter Name	Parameter Value
requestId	25:7164
payload	Not applicable
Event	Not applicable
eventID	996c7386-8ce8-4e4e-803d-993064a287c7
sequenceNumber	28
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	CTIController
subscriptionId	80208e52-c673-4041-a954-0a1d91b72802
channelId	5616863d-9336-4296-921a-1fb1fce69705
targetId	ACD_CTI1@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-53197:0
extTrackingId	94:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271712495635
removeTime	1271712593257
bounced	Not applicable
acdName	ACD_CTI1_Name
acdNumber	tel:+15006001025
acdPriority	0-Highest
addTimeInPriority Bucket	1271712495635

F14 EventResponse

Parameter Name	Parameter Value
requestId	25:7164
payload	Not applicable
EventResponse	Not applicable
eventID	996c7386-8ce8-4e4e-803d-993064a287c7
statusCode	200
reason	OK

9.4.3 Call Overflow

This scenario illustrates a case where a call is overflowed because it is not handled by an agent within the amount of time specified by the overflow policy.

In this scenario, network party N1 is initially connected to the ACD1 via call C1. Call C1 is maintained in the ACD1 queue. The overflow policy is configured to send the call to ACD2.

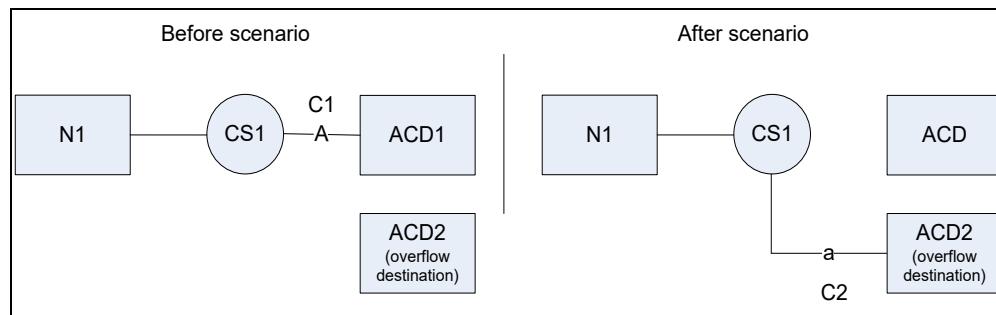


Figure 149 Call Overflow Message Flow

Activity	Monitored ACD1	Monitored ACD2	Comments
The call has been in the queue for longer than the value specified by the overflow policy.	ACD Call overflowed Event (F1) Response (F3)		
Indication that the call is added to the ACD2 queue.		ACD Call Added Event (F3) Response (F4)	

F1 ACDCallOverflowedEvent

Parameter Name	Parameter Value
requestId	4:15
payload	Not applicable
Event	Not applicable
eventID	3e1f6a19-5861-41f5-9342-99d2d9fb9938
sequenceNumber	3
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	1f0bad27-2945-45c3-acfc-556e20b5d931
channelId	f4e45903-2b01-4cf8-86bd-56a73d674560
targetId	ACD_CTL1@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-53349:0
extTrackingId	103:1



Parameter Name	Parameter Value
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271713802660
removeTime	1271713812670
acdName	ACD_CTI1_Name
acdNumber	tel:+15006001025
acdPriority	0-Highest
addTimeInPriorityBucket	1271713802660
overflowReason	time
redirect	Not applicable
address	tel:1026
reason	deflection
redirectTime	1271713812671

F2 EventResponse

Parameter Name	Parameter Value
requestId	4:15
payload	Not applicable
EventResponse	Not applicable
eventID	3e1f6a19-5861-41f5-9342-99d2d9fb9938
statusCode	200
reason	OK

F3 ACDCallAddedEvent

Parameter Name	Parameter Value
requestId	4:16
payload	Not applicable
Event	Not applicable
eventID	f44140b2-1cbb-457f-8b82-b7d8bce51fa1
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	37070686-e5c6-4200-a040-21b37c660390
channelId	f4e45903-2b01-4cf8-86bd-56a73d674560

Parameter Name	Parameter Value
targetId	ACD_CTI2@broadsoft.com
eventData	Not applicable
queueEntry	Not applicable
callId	callhalf-53353:0
extTrackingId	103:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271713812744
preservedWaitTime	10010
acdName	ACD_CTI2_Name
acdNumber	tel:+15006001026
acdPriority	0-Highest
addTimeInPriorityBucket	1271713812744
preservedWaitTimeln PriorityBucket	10010
position	1

F4 EventResponse

Parameter Name	Parameter Value
requestId	4:16
payload	Not applicable
EventResponse	Not applicable
eventId	f44140b2-1cbb-457f-8b82-b7d8bce51fa1
statusCode	200
reason	OK

9.4.4 Call Forwarded

This scenario illustrates a case where a call is forwarded because of the Forced Forwarding policy. In this scenario, network party N1 initiates a call toward the ACD for which the Forced Forwarding policy is enabled to subscriber S1.

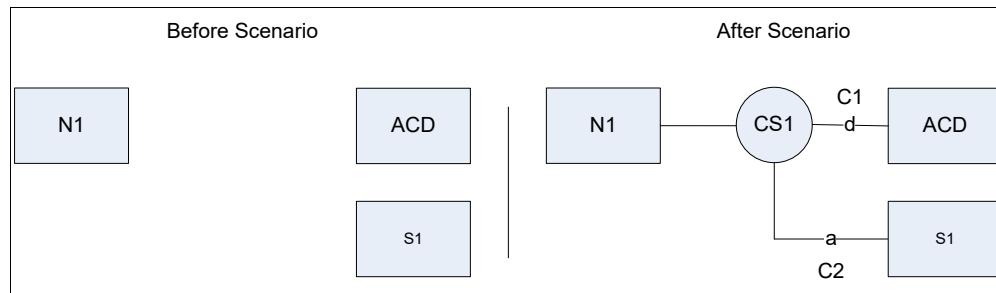


Figure 150 Call Forwarded Message Flow

Activity	Monitored ACD	Monitored S1	Comments
N1 dial ACD phone number. The call is automatically forwarded toward destination S1.	ACD Call Forwarded Event (F1) Response (F2)	Call Received Event (F3) Response (F4)	The Call Received event contains the detail of the redirection performed.

F1 ACDCallForwardedEvent

Parameter Name	Parameter Value
requestId	48:1960
payload	Not applicable
Event	Not applicable
eventID	56a625ea-45a4-4e50-a741-6061d35f8ada
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	CTIController
subscriptionId	5f496571-8177-4257-b303-b14909ae4966
channelId	9488b2a7-abb4-4dd3-b0f4-0ea7ea4ce0e2
targetId	ACD_CTI1@broadsoft.com
eventData	Not applicable
callId	callhalf-78933:0
extTrackingId	332:1
acdName	ACD_CTI1_Name
acdNumber	tel:+15006001025
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333



Parameter Name	Parameter Value
callType	Network
redirect	Not applicable
address	tel:1011
reason	deflection
redirectTime	1271882708100

F2 EventResponse

Parameter Name	Parameter Value
requestId	48:1960
payload	Not applicable
EventResponse	Not applicable
eventID	56a625ea-45a4-4e50-a741-6061d35f8ada
statusCode	200
reason	OK

F3 CallReceivedEvent

Parameter Name	Parameter Value
requestId	1:2113
payload	Not applicable
Event	Not applicable
eventID	aefd7f81-e422-4997-b0b4-13f727a8840a
sequenceNumber	5
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	0893a40a-c708-4af8-9ebf-e3a299913b46
channelId	e89e1bb5-5223-4af1-849d-0e468ee9c07c
targetId	subscriberS1@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-79433:0
extTrackingId	332:1
personality	Terminator
state	Alerting
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333

Parameter Name	Parameter Value
callType	Network
redirections	Not applicable
redirection	Not applicable
party	Not applicable
name	ACD_CTI1_FirstName ACD_CTI1_LastName
address	tel:+15006001025
userId	ACD_CTI1@broadsoft.com
callType	Group
reason	deflection
startTime	1271882708100

F4 EventResponse

Parameter Name	Parameter Value
requestId	1:2113
payload	Not applicable
EventResponse	Not applicable
eventID	aefd7f81-e422-4997-b0b4-13f727a8840a
statusCode	200
reason	OK

9.4.5 Holiday Policy Applied

This scenario illustrates a case where a call is forwarded because of the Holiday policy. In this scenario, network party N1 initiates a call toward the ACD for which the Holiday policy is enabled and forwarding to subscriberS1.

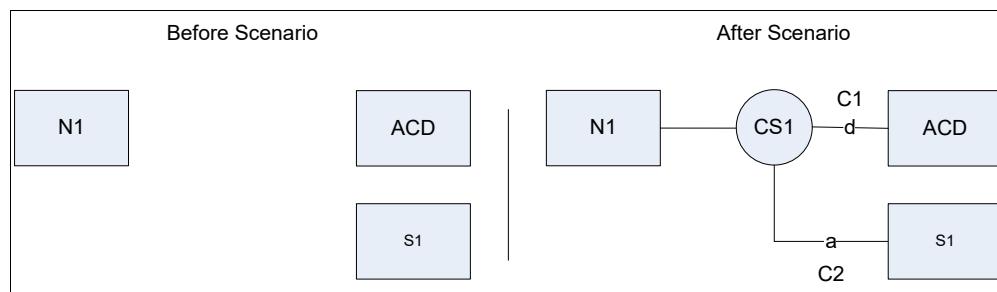


Figure 151 Holiday Policy Applied Message Flow



Activity	Monitored ACD	Monitored S1	Comments
N1 dial ACD phone number. The call is automatically forwarded toward destination S1 because of the holiday policy.	ACD Holiday Policy Applied Event (F1) Response (F2)	Call Received Event (F3) Response (F4)	The Call Received event contains the detail of the redirection performed.

F1 ACDHolidayPolicyAppliedEvent

Parameter Name	Parameter Value
requestId	1:2075
payload	Not applicable
Event	Not applicable
eventID	4ad5f5c0-0ce6-4c6d-b9c3-bbe8eb883e8a
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com
External ApplicationId	AppCtlId
subscriptionId	c09f2dee-32ab-4408-9ea9-cb9806331191
channelId	e89e1bb5-5223-4af1-849d-0e468ee9c07c
targetId	ACD_CTL1@broadsoft.com
eventData	Not applicable
callId	callhalf-79297:0
extTrackingId	70:1
acdName	ACD_CTL1_Name
acdNumber	tel:+15006001025
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
redirect	Not applicable
address	tel:1011
reason	deflection
redirectTime	1272036730325

F2 EventResponse

Parameter Name	Parameter Value
requestId	1:2075
payload	Not applicable
EventResponse	Not applicable



Parameter Name	Parameter Value
eventID	4ad5f5c0-0ce6-4c6d-b9c3-bbe8eb883e8a
statusCode	200
reason	OK

F3 CallReceivedEvent

Parameter Name	Parameter Value
requestId	0:7696
payload	Not applicable
Event	Not applicable
eventID	e4d7bb7a-d046-4e3b-96f8-9d91ac2a30e0
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com
External ApplicationId	AppCtlId
subscriptionId	2a435cb9-05ef-4673-8692-46e0ec91106f
channelId	da60ef24-d1af-44bb-afca-3dc9c77879da
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-107053:0
extTrackingId	70:1
personality	Terminator
state	Alerting
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
userId	NetworkN1@broadsoft.com
callType	Enterprise
redirections	Not applicable
redirection	Not applicable
party	Not applicable
name	ACD_CTI1_FirstName ACD_CTI1_LastName
address	tel:+15006001025
userId	ACD_CTI1@broadsoft.com
callType	Group
reason	deflection

Parameter Name	Parameter Value
startTime	1272036730325

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:7696
payload	Not applicable
EventResponse	Not applicable
eventID	e4d7bb7a-d046-4e3b-96f8-9d91ac2a30e0
statusCode	200
reason	OK

9.4.6 Call Stranded

This scenario illustrates a case where a call is stranded because the last agent sign-out of the queue. In this scenario, network party N1 is initially connected to the ACD via call C1. Call C1 is maintained in the ACD queue.

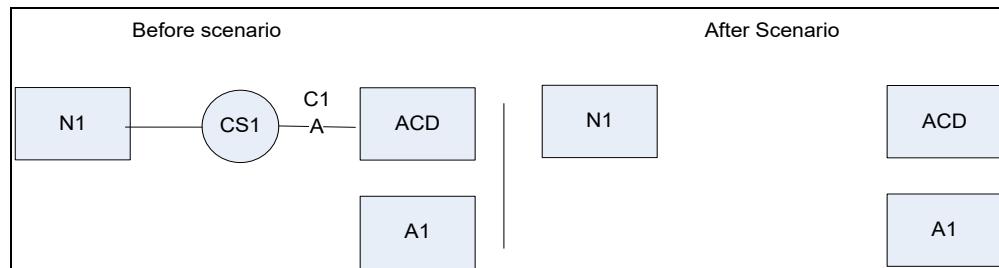


Figure 152 Call Stranded Message Flow

Activity	Monitored ACD	Monitored A1	Comments
The last agent assigned to this ACD sets its state <i>Sign-out</i> .		Agent State Changed Event (F1) Response (F2)	
The call is removed from the queue and an announcement in a loop is played to the caller.	ACD Call Stranded Event (F3) Response (F4)		

F1 AgentStateEvent

Parameter Name	Parameter Value
requestId	0:105
payload	Not applicable
Event	Not applicable
eventID	ee5b31d8-2ed4-4d70-9b12-f73aefcc83e1
sequenceNumber	4



Parameter Name	Parameter Value
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	edd668c9-3d8f-4ca9-a613-25c8c7908004
channelId	2c43bc73-962e-48c8-a324-ad7fba0c4e43
targetId	agentA1@broadsoft.com
eventData	Not applicable
agentStateInfo	Not applicable
state	Sign-out
stateTimestamp	1271724227541

F2 EventResponse

Parameter Name	Parameter Value
requestId	0:105
payload	Not applicable
EventResponse	Not applicable
eventID	ee5b31d8-2ed4-4d70-9b12-f73aefcc83e1
statusCode	200
reason	OK

F3 ACDCallStrandedEvent

Parameter Name	Parameter Value
requestId	0:106
payload	Not applicable
Event	Not applicable
eventID	6f0b5bec-a589-45ba-8d05-e3ddcd37271e
sequenceNumber	13
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	38d2428a-37b7-4d9a-8f95-08205b7d9147
channelId	2c43bc73-962e-48c8-a324-ad7fba0c4e43
targetId	ACD_CTL1@broadsoft.com
eventData	Not applicable
queueEntry	Not applicable
callId	callhalf-54545:0
extTrackingId	176:1
remoteParty	Not applicable

Parameter Name	Parameter Value
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271724216743
removeTime	1271724227613
acdName	ACD_CTI1_Name
acdNumber	tel:+15006001025
acdPriority	0-Highest
addTimeInPriority Bucket	1271724216742

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:106
payload	Not applicable
EventResponse	Not applicable
eventID	6f0b5bec-a589-45ba-8d05-e3ddcd37271e
statusCode	200
reason	OK

9.4.7 Call Transfer

This scenario illustrates a successful ACD Blind Transfer. Network party N1 is connected to the ACD via call C1 when the remote application initiates a blind transfer to subscriber S1. In this scenario, subscriber S1 is available and valid, and the call is answered by subscriber S1.

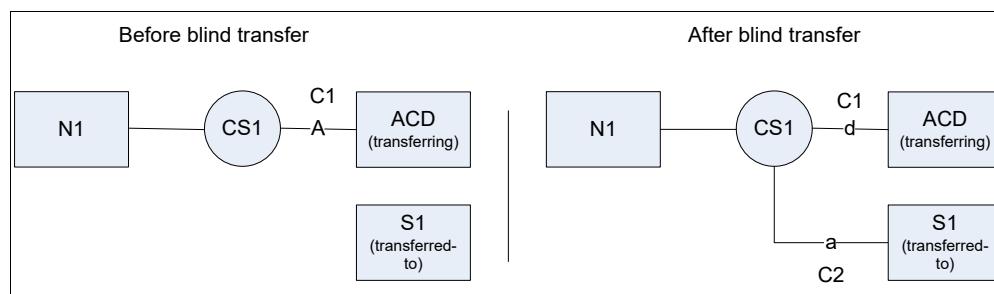


Figure 153 Call Transfer Message Flow

Activity	Monitored ACD	Monitored Subscriber S1	Comments
A Transfer Request to subscriber S1 is invoked on behalf of ACD.	Request: ACD Blind Transfer (F1) Response (F2)		



Activity	Monitored ACD	Monitored Subscriber S1	Comments
Indication that a transfer has been initiated from the ACD. Subscriber S1 is alerted.	ACD Call Transferred Event (F3) Response (F4)	Call Received Event (F5) Response (F6)	The Call Transferred event specifies that the call is now in <i>Alerting</i> state and remoteParty is now S1. The Call Received event contains detail on the redirection that was performed.

F1 Request

Parameter Name	Parameter Value
requestId	9
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
ACDBlindTransferRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/callcenter/%callcenterId%/calls/%callId%/blindtransfer?%phono%
method	PUT
version	Not applicable
params	Not applicable
phoneno	6001011
callId	callhalf-58225:0
callcenterId	ACD_CTI1@broadsoft.com
payload	Not applicable

F2 Response

Parameter Name	Parameter Value
requestId	9
sessionId	103
statusCode	200
reason	Not applicable
payload	Not applicable

F3 ACDCallTransferredEvent

Parameter Name	Parameter Value
requestId	0:7586
payload	Not applicable
Event	Not applicable
eventide	d9eec76e-33c1-4de7-9e63-0b82d02b85cd
sequenceNumber	15
userId	adminCTI@broadsoft.com@broadsoft.com



Parameter Name	Parameter Value
externalApplicationId	AppCtlId
subscriptionId	fab8103d-d7b9-4be7-8cd2-8bc711e8d7ac
channelId	367062da-6573-461e-b5df-595734a68e00
targetId	ACD_CTI1@broadsoft.com
eventData	Not applicable
queueEntry	Not applicable
callId	callhalf-58225:0
extTrackingId	255:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271770544207
removeTime	1271770576187
acdName	ACD_CTI1_Name
acdNumber	tel:+15006001025
acdPriority	0-Highest
addTimeInPriority Bucket	1271770544207
redirect	Not applicable
address	tel:6001011
reason	transfer
redirectTime	1271770576187

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:7586
payload	Not applicable
EventResponse	Not applicable
eventide	d9eec76e-33c1-4de7-9e63-0b82d02b85cd
statusCode	200
reason	OK

F5 CallReceivedEvent

Parameter Name	Parameter Value
requestId	0:7587
payload	Not applicable

Parameter Name	Parameter Value
Event	Not applicable
eventID	107a850c-eb8b-4403-abd5-2e6fa55a004f
sequenceNumber	12
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	0dcb7fa4-d838-4ce8-8120-5e7f4faa2e8c
channelId	367062da-6573-461e-b5df-595734a68e00
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-58237:0
extTrackingId	255:1
networkCallId	BW093616220200410198054573@192.168.13.180
personality	Terminator
state	Alerting
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
redirections	Not applicable
redirection	Not applicable
party	Not applicable
name	ACD_CTI1_FirstName ACD_CTI1_LastName
address	tel:+15006001025
userId	ACD_CTI1@broadsoft.com
callType	Group
reason	deflection
startTime	1271770576221

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:7587
payload	Not applicable
EventResponse	Not applicable
eventID	107a850c-eb8b-4403-abd5-2e6fa55a004f
statusCode	200

Parameter Name	Parameter Value
reason	OK

9.4.8 Call Updated

This scenario illustrates the case where a Call Updated event is generated. Network party N1 is connected to the ACD via call C1 when the N1 transfers the call to destination N2.

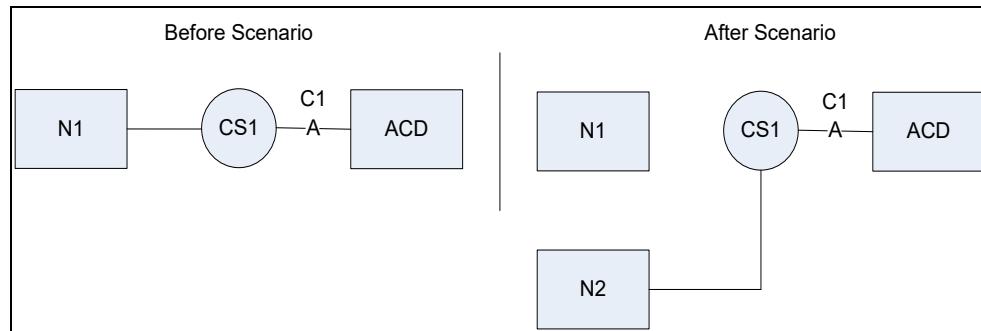


Figure 154 Call Updated Message Flow

Activity	Monitored ACD	Comments
A Transfer to Network Party N2 is performed by N1.		
Indication that the remote party has changed.	ACD Call Updated Event (F1) Response (F2)	The Call Updated event specifies that remoteParty is now N2.

F1 ACDCallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:7738
payload	Not applicable
Event	Not applicable
eventID	b0f9a450-9a29-4691-a3ec-6c8de2c277cd
sequenceNumber	19
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	fab8103d-d7b9-4be7-8cd2-8bc711e8d7ac
channelId	367062da-6573-461e-b5df-595734a68e00
targetId	ACD_CTL1@broadsoft.com
eventData	Not applicable
queueEntry	Not applicable
callId	callhalf-58365:0

Parameter Name	Parameter Value
extTrackingId	262:1
remoteParty	Not applicable
address	tel:+17003334
callType	Network
addTime	1271771165048
acdName	ACD_CTI1_Name
acdNumber	tel:+15006001025
acdPriority	0-Highest
addTimeInPriority Bucket	1271771165048
reason	Call Transferred

F2 EventResponse

Parameter Name	Parameter Value
requestId	0:7738
payload	Not applicable
EventResponse	Not applicable
eventID	b0f9a450-9a29-4691-a3ec-6c8de2c277cd
statusCode	200
reason	OK

9.4.9 Call Promoted, Reordered and Finally Escaped

This scenario illustrates a case where a call is manually promoted, then reordered and is finally released by the caller. In this scenario, network party N1 initiates a call toward the ACD.

In this scenario, network party N1 is initially connected to the ACD via call C1. Call C1 is maintained in the ACD queue at position 1 with priority 1-High.

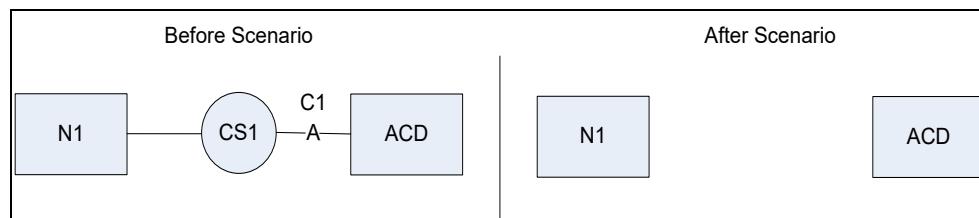


Figure 155 Call Promoted, Reordered, and Finally Escaped Message Flow

Activity	Monitored ACD	Comments
A remote application performs an ACD Promote Call to priority 0-Highest in an attempt to speed up the distribution of the call.	ACD Promote Call Request(F1) Response (F2)	The Request response specifies that the position is now at 2.

Activity	Monitored ACD	Comments
Indication that the call was promoted to position 2 with priority 0-Highest.	ACD Call Promoted Event (F3) Response (F4)	
After several minutes, a remote application performs an ACD Call Reorder to position 1 in an attempt to speed up the distribution of the call.	ACD Reorder Call Request (F5) Response (F6)	The Request response specifies that the position is now at 1.
Indication that the call was promoted to position 1.	ACD Call Reordered Event (F7) Response (F8)	
The caller decides to leave the queue by entering the configured escaped digit.	ACD Call Escaped Event (F9) Response (F10)	The call is removed from the queue and provided with busy call handling.

F1 Request

Parameter Name	Parameter Value
requestId	6
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
ACDPromoteCallRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/callcenter/%subscriberId%/calls/%callId%/promote
method	PUT
version	Not applicable
params	Not applicable
callId	callhalf-59645:0
subscriberId	ACD_CTI1@broadsoft.com
payload	Not applicable
ACDPromote	Not applicable
priority	0-Highest

F2 Response

Parameter Name	Parameter Value
requestId	6
sessionId	194
statusCode	200
reason	OK
payload	Not applicable
QueueCallOrder	Not applicable
position	2

F3 ACDCallPromotedEvent

Parameter Name	Parameter Value
requestId	4:9272
payload	Not applicable
Event	Not applicable
eventID	629625f5-f033-400d-884e-3a99af7a62c6
sequenceNumber	4
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	7573e2a2-7287-4352-8940-7c04b526dc98
channelId	bdfe939b-fbe1-425e-9370-c9b81097b141
targetId	ACD_CTI1@broadsoft.com
EventData	Not applicable
queueEntry	Not applicable
callId	callhalf-59645:0
extTrackingId	354:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271776555321
acdName	ACD_CTI1_2_Name
acdNumber	tel:+15006001027
acdPriority	0-Highest
addTimeInPriority Bucket	1271776586140
position	2

F4 EventResponse

Parameter Name	Parameter Value
requestId	4:9272
payload	Not applicable
EventResponse	Not applicable
eventID	629625f5-f033-400d-884e-3a99af7a62c6
statusCode	200
reason	OK



F5 Request

Parameter Name	Parameter Value
requestId	1
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
ACDReorderCall Request	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/callcenter/%subscriberId%/calls/%callId%/reorder?%index%
method	PUT
version	Not applicable
params	Not applicable
index	1
callId	callhalf-59645:0
subscriberId	ACD_CTI1@broadsoft.com
payload	Not applicable

F6 Response

Parameter Name	Parameter Value
requestId	1
sessionId	207
statusCode	200
reason	OK
payload	Not applicable
QueueCallOrder	Not applicable
position	1

F7 ACDCallReorderedEvent

Parameter Name	Parameter Value
requestId	4:9404
payload	Not applicable
Event	Not applicable
eventID	101cfad3-1e41-45ed-a185-22aaa26fb078
sequenceNumber	5
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	7573e2a2-7287-4352-8940-7c04b526dc98
channelId	bdfe939b-fbe1-425e-9370-c9b81097b141

Parameter Name	Parameter Value
targetId	ACD_CTI1@broadsoft.com
eventData	Not applicable
queueEntry	Not applicable
callId	callhalf-59645:0
extTrackingId	354:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271776555321
reordered	Not applicable
acdName	ACD_CTI1_2_Name
acdNumber	tel:+15006001027
acdPriority	0-Highest
addTimeInPriority Bucket	1271776586140
position	1

F8 EventResponse

Parameter Name	Parameter Value
requestId	4:9404
payload	Not applicable
EventResponse	Not applicable
eventID	101cfad3-1e41-45ed-a185-22aaa26fb078
statusCode	200
reason	OK

F9 ACDCallEscapedEvent

Parameter Name	Parameter Value
requestId	4:9408
payload	Not applicable
Event	Not applicable
eventID	ff9aafb7-41b2-4297-b723-5b1af8635dc3
sequenceNumber	6
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	7573e2a2-7287-4352-8940-7c04b526dc98



Parameter Name	Parameter Value
channelId	befe939b-fbe1-425e-9370-c9b81097b141
targetId	ACD_CTI1@broadsoft.com
eventData	Not applicable
queueEntry	Not applicable
callId	callhalf-59645:0
extTrackingId	354:1
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
addTime	1271776555321
removeTime	1271777047356
reordered	Not applicable
acdName	ACD_CTI1_2_Name
acdNumber	tel:+15006001027
acdPriority	0-Highest
addTimeInPriority Bucket	1271776586140

F10 EventResponse

Parameter Name	Parameter Value
requestId	4:9408
payload	Not applicable
EventResponse	Not applicable
eventId	ff9aafb7-41b2-4297-b723-5b1af8635dc3
statusCode	200
reason	OK

9.5 Agent Message Flows (AS only)

The next subsections illustrate the following scenarios:

- Agent State change
- Escalate to Supervisor
- Emergency call to Supervisor



9.5.1 Agent Join and State Change

In this scenario, an agent joins the ACD. It then changes its state from *Unavailable* to *Available*.

Activity	Monitored Agent A1	Comments
The agent joins the ACD.	Agent Join Update Event (F1) Response (F2)	
The agent is in state unavailable. A Set Agent State Request is performed on behalf of agent A1.	Set Agent State Request (F3) Response (F4)	
The agent goes from <i>Unavailable</i> to <i>Available</i> .	Agent State Event (F5) Response (F6)	

F1 ACDAgentJoinUpdateEvent

Parameter Name	Parameter Value
requestId	0:132
payload	Not applicable
Event	Not applicable
eventID	4e819eea-d91c-47a5-b040-d9f85400ee86
sequenceNumber	7
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	9f35d69c-bc8c-4dac-91af-f94766e35643
channelId	5e565648-7293-4c0b-ba8e-5d8608a1d466
targetId	agentA1@broadsoft.com
EventData	Not applicable
ccAgentJoinUpdateData	Not applicable
joinInfo	Not applicable
acdUserId	ACD_CTL1@broadsoft.com
joined	Not applicable

F2 EventResponse

Parameter Name	Parameter Value
requestId	0:132
payload	Not applicable
EventResponse	Not applicable
eventID	4e819eea-d91c-47a5-b040-d9f85400ee86
statusCode	200
reason	OK

F3 Request

Parameter Name	Parameter Value
requestId	3
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
SetAgentState Request	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/services/callcenter
method	PUT
version	Not applicable
params	Not applicable
subscriberId	agentA1@broadsoft.com
payload	Not applicable
CallCenter	Not applicable
agentACDState	Available

F4 Response

Parameter Name	Parameter Value
requestId	3
sessionId	45
statusCode	200
reason	OK
payload	Not applicable

F5 AgentStateEvent

Parameter Name	Parameter Value
requestId	0:133
payload	Not applicable
Event	Not applicable
eventID	217bc985-ca4b-4ea8-85b1-adf1f6a254b6
sequenceNumber	8
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	9f35d69c-bc8c-4dac-91af-f94766e35643
channelId	5e565648-7293-4c0b-ba8e-5d8608a1d466
targetId	agentA1@broadsoft.com
eventData	Not applicable
agentStateInfo	Not applicable

Parameter Name	Parameter Value
state	Available
stateTimestamp	Not applicable
value	1271786934182
signInTimestamp	Not applicable
value	1271786534182
totalAvailableTime	0

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:133
payload	Not applicable
EventResponse	Not applicable
eventID	217bc985-ca4b-4ea8-85b1-adf1f6a254b6
statusCode	200
reason	OK

9.5.2 Escalate to Supervisor

This scenario illustrates a successful Escalate to Supervisor Request on behalf of agent A1. In this scenario, agent A1 is connected to network user N1 via call session CS1. Agent A1 is assigned to supervisor S1 who is available to handle the escalation call.

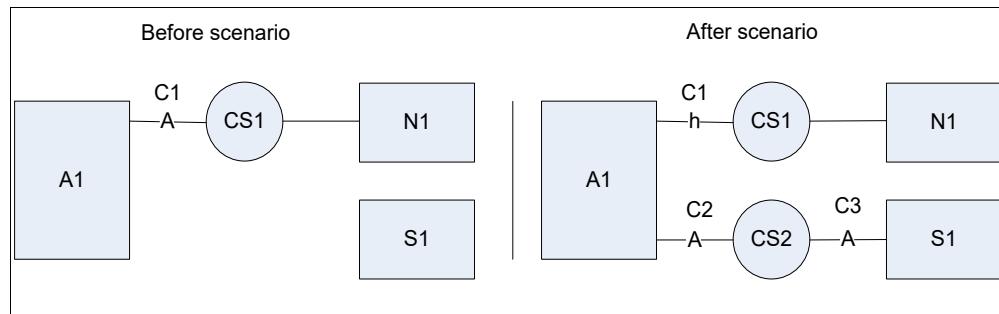


Figure 156 Call Promoted, Reordered, and Finally Escaped Message Flow

Activity	Monitored Subscriber A1	Monitored Subscriber S1	Comments
An Escalate to Supervisor Request is invoked on behalf of agent A1.	Escalate to Supervisor Request (F1) Response (F2)		The Escalate to Supervisor Request does not specify any supervisor. Cisco BroadWorks therefore identifies S1 as the available supervisor.
Indication that the service has been initiated from this subscriber.	Call Originating Event (F3) Response (F4)		The “Click-to-Dial” personality indicates that the agent A1 is being prompted (via ringing, for example) to go off-hook.



Activity	Monitored Subscriber A1	Monitored Subscriber S1	Comments
Indicates that call C1 is being held.	Call Held Event (F5) Response (F6)		
Agent A1 goes off-hook and is connected in the call.	Call Originated Event (F7) Response (F8)		This Alerting is different from the previous one as the personality is 'Originator'
Supervisor S1 endpoint begins to ring and A1 receives ringing tone.		Call Received Event (F9) Response (F10)	
Supervisor S1 answers the call by manually going off-hook.	Call Answered Event (F15) Response (F16)	Hook Status Event (F11) Response (F12) Call Answered Event (F13) Response (F14)	Active for both users. Agent and supervisor are speaking.

F1 Request

Parameter Name	Parameter Value
requestId	13
sessionId	27
credentials	YXNvdXRoQG10bGFzZGV2ODcubmV0Om10bGxhYg==
EscalateToSupervisorRequest	N/A
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/escalate?%address%&%location%
method	POST
version	N/A
params	N/A
location	All
address	tel:1011
subscriberId	agentA1@broadworks.com
payload	N/A

F2 Response

Parameter Name	Parameter Value
requestId	13
sessionId	27
statusCode	201
reason	Created
payload	N/A
CallStartInfo	N/A
callId	callhalf-29445:1
externalTrackingId	329e389e-a78c-478e-9215-b66acf499528

F3 CallOriginatingEvent

Parameter Name	Parameter Value
requestId	0:7699
payload	N/A
Event	N/A
eventID	5142261c-4cbc-415b-a668-be9895bf4f16
sequenceNumber	53
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	eb134d07-4278-4a86-b499-b777c0e41c9b
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	agentA1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-29445:1
extTrackingId	329e389e-a78c-478e-9215-b66acf499528
personality	Click-to-Dial
state	Alerting
remoteParty	N/A
address	tel:#831011
callType	Unknown
appearance	2
startTime	1382538630521

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:7699
payload	N/A
EventResponse	N/A
eventID	5142261c-4cbc-415b-a668-be9895bf4f16
statusCode	200
reason	OK

F5 CallHeldEvent

Parameter Name	Parameter Value
requestId	0:7700
payload	N/A



Parameter Name	Parameter Value
Event	N/A
eventID	0445a8cc-07d5-47ee-ac56-f51480587510
sequenceNumber	54
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	eb134d07-4278-4a86-b499-b777c0e41c9b
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	agentA1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-29445:0
extTrackingId	034c31a7-b4cb-406e-8214-06411c68fc81
personality	Terminator
state	Held
remoteParty	N/A
name	5007003333
address	tel:+15007003333
callType	Network
endpoint	N/A
addressOfRecord	5006001020@broadworks.com
appearance	1
startTime	1382538618443
answerTime	1382538619887
heldTime	1382538631811
acdCallInfo	N/A
acdUserId	ACD_CTI1@broadworks.com
acdName	ACDName
acdNumber	tel:+15006001025
numCallsInQueue	0
waitTime	0
callingPartyInfo	N/A
name	5007003333
address	tel:+15007003333
callType	Network



F6 EventResponse

Parameter Name	Parameter Value
requestId	0:7700
payload	N/A
EventResponse	N/A
eventID	0445a8cc-07d5-47ee-ac56-f51480587510
statusCode	200
reason	OK

F7 CallOriginatedEvent

Parameter Name	Parameter Value
requestId	0:7701
payload	N/A
Event	N/A
eventID	52c8b778-1d12-4906-ab84-f6a6bbf95764
sequenceNumber	55
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	eb134d07-4278-4a86-b499-b777c0e41c9b
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	agentA1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-29445:1
extTrackingId	329e389e-a78c-478e-9215-b66acf499528
personality	Originator
state	Alerting
remoteParty	N/A
name	subscriberS1Firstname subscriberS1Lastname
address	tel:1011
userId	subscriberS1@broadworks.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	N/A
addressOfRecord	5006001020@broadworks.com
appearance	2
startTime	1382538631823



Parameter Name	Parameter Value
agentEscalation	Normal

F8 EventResponse

Parameter Name	Parameter Value
requestId	0:7701
payload	N/A
EventResponse	N/A
eventID	52c8b778-1d12-4906-ab84-f6a6bbf95764
statusCode	200
reason	OK

F9 CallReceivedEvent

Parameter Name	Parameter Value
requestId	0:7702
payload	N/A
Event	N/A
eventID	ffe90e48-679f-422c-aa27-2430cfb293b4
sequenceNumber	2
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	7a8eb65c-3e33-4260-a060-48c5129f000d
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	subscriberS1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-29449:0
extTrackingId	329e389e-a78c-478e-9215-b66acf499528
personality	Terminator
state	Alerting
remoteParty	N/A
name	agentA1Firstname agentA1Lastname
address	tel:1020
userId	agentA1@broadworks.com
userDN	tel:+15006001020;ext=1020
callType	Group
startTime	1382538631823



Parameter Name	Parameter Value
agentEscalation	Normal

F10 EventResponse

Parameter Name	Parameter Value
requestId	0:7702
payload	N/A
EventResponse	N/A
eventID	ffe90e48-679f-422c-aa27-2430cfb293b4
statusCode	200
reason	OK

F11 HookStatusEvent

Parameter Name	Parameter Value
requestId	0:7704
payload	N/A
Event	N/A
eventID	f84bce5c-373f-4aa5-b8b7-89793ef4698b
sequenceNumber	4
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	7a8eb65c-3e33-4260-a060-48c5129f000d
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	subscriberS1@broadworks.com
EventData	N/A
hookStatus	Off-Hook

F12 EventResponse

Parameter Name	Parameter Value
requestId	0:7704
payload	N/A
EventResponse	N/A
eventID	f84bce5c-373f-4aa5-b8b7-89793ef4698b
statusCode	200
reason	OK



F13 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:7705
payload	N/A
Event	N/A
eventID	9b28cdc3-b4cb-4db6-b47b-f1426dfa817e
sequenceNumber	56
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	eb134d07-4278-4a86-b499-b777c0e41c9b
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	agentA1@broadworks.com
EventData	N/A
call	N/A
callId	callhalf-29445:1
extTrackingId	329e389e-a78c-478e-9215-b66acf499528
personality	Originator
state	Active
remoteParty	N/A
name	subscriberS1Firstname subscriberS1Lastname
address	tel:1011
userId	subscriberS1@broadworks.com
userDN	tel:+15006001011;ext=1011
callType	Group
endpoint	N/A
addressOfRecord	5006001020@broadworks.com
appearance	2
startTime	1382538631823
answerTime	1382538639920
agentEscalation	Normal

F14 EventResponse

Parameter Name	Parameter Value
requestId	0:7705
payload	N/A
EventResponse	N/A
eventID	9b28cdc3-b4cb-4db6-b47b-f1426dfa817e



Parameter Name	Parameter Value
statusCode	200
reason	OK

F15 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:7706
payload	N/A
Event	N/A
eventID	65e3f25a-c4ce-4d8f-a09c-9eb905ae1cae
sequenceNumber	5
userId	asouth@broadworks.com
externalApplicationId	com.broadsoft.remoteapp.routepoint
subscriptionId	7a8eb65c-3e33-4260-a060-48c5129f000d
channelId	0a41fe2f-7e46-4bf2-b1cc-29247fa6dfa3
targetId	subscriberS1@broadworks.com
eventData	N/A
call	N/A
callId	callhalf-29449:0
extTrackingId	329e389e-a78c-478e-9215-b66acf499528
personality	Terminator
state	Active
remoteParty	N/A
name	agentA1Firstname agentA1Lastname
address	tel:1020
userId	agentA1@broadworks.com
userDN	tel:+15006001020;ext=1020
callType	Group
endpoint	N/A
addressOfRecord	5006001011@broadworks.com
appearance	1
startTime	1382538631823
answerTime	1382538639920
agentEscalation	Normal

F16 EventResponse

Parameter Name	Parameter Value
requestId	0:7706
payload	N/A
EventResponse	N/A
eventID	65e3f25a-c4ce-4d8f-a09c-9eb905ae1cae
statusCode	200
reason	OK

9.5.3 Emergency Call to Supervisor

This scenario illustrates a successful Emergency call to Supervisor Request on behalf of agent A1. In this scenario, agent A1 is connected to network party N via call session CS1. Agent A1 is assigned to supervisor S1 who is available to handle the emergency call.

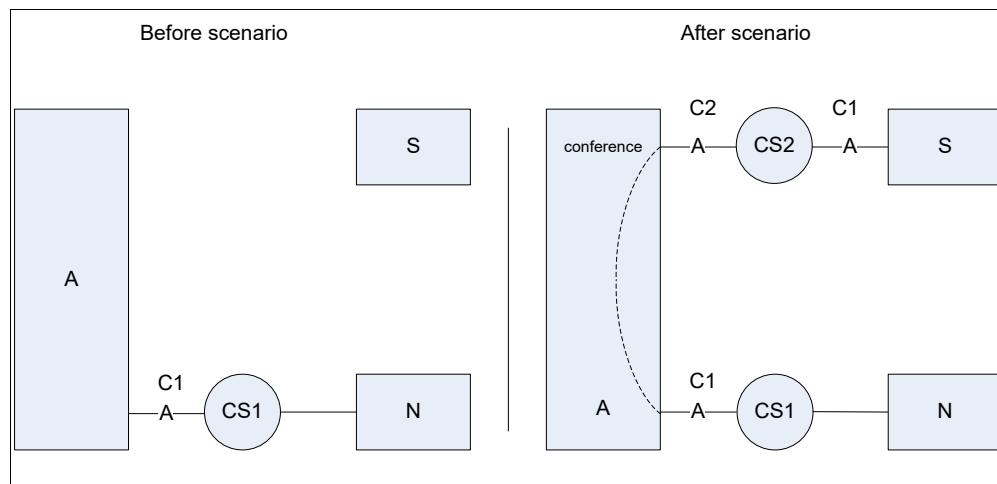


Figure 157 Emergency Call to Supervisor Message Flow

Activity	Monitored Agent A1	Monitored Supervisor S1	Comments
An Escalate to Supervisor Request is invoked on behalf of agent A1.	Escalate to Supervisor Request (F1) Response (F2)		The Escalate to Supervisor Request does not specify any supervisor. Cisco BroadWorks therefore identifies S1 as the available supervisor.
Indication that call C2 is initiated.	Call Originated Event (F3) Response (F4)		
Supervisor S1 endpoint begins to ring.		Call Received Event (F5) Response (F6)	
Supervisor S1 answers the call by manually going off-hook.	Call Answered Event (F11) Response (F12)	Hook Status Event (F7) Response (F8) Call Answered Event (F9) Response (F10)	

Activity	Monitored Agent A1	Monitored Supervisor S1	Comments
The conference is established.	Call Updated Event (F13) Response (F14) Conference Started Event (F15) Response (F16)		The Call Updated event report there is no endpoint associated with call C1.

F1 Request

Parameter Name	Parameter Value
requestId	5
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
EmergencyCallTo SupervisorRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/%callId%/emergencyescalate?%address%
method	POST
version	Not applicable
params	Not applicable
address	1011
callId	callhalf-62131:0
subscriberId	agentA1@broadsoft.com
payload	Not applicable

F2 Response

Parameter Name	Parameter Value
requestId	5
sessionId	87
statusCode	201
reason	Created
payload	Not applicable
CallStartInfo	Not applicable
callId	callhalf-62131:1
externalTrackingId	446:1

F3 CallOriginatedEvent

Parameter Name	Parameter Value
requestId	0:249
payload	Not applicable
Event	Not applicable



Parameter Name	Parameter Value
eventID	0af23481-1bcc-49d1-be35-c11506c0f2fb
sequenceNumber	30
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	f64ffa17-5341-43ad-81a3-a73113f170da
channelId	712398d4-0cd9-4363-b53c-cde224816059
targetId	agentA1@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-62131:1
extTrackingId	446:1
personality	Originator
state	Alerting
remoteParty	Not applicable
name	subscriberS1FirstName subscriberS1LastName
address	tel:1011
userId	subscriberS1@broadsoft.com
userDN	tel:+15006001011;ext=1011
callType	Group
diversionInhibited	Not applicable
startTime	1271789995718
agentEscalation	Emergency

F4 EventResponse

Parameter Name	Parameter Value
requestId	0:249
payload	Not applicable
EventResponse	Not applicable
eventID	0af23481-1bcc-49d1-be35-c11506c0f2fb
statusCode	200
reason	OK

F5 CallReceivedEvent

Parameter Name	Parameter Value
requestId	0:250
payload	Not applicable



Parameter Name	Parameter Value
Event	Not applicable
eventID	2242376b-d19a-4ce1-bdff-64b85aab500b
sequenceNumber	8
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	573c743f-5561-4748-b46f-4d1fe1b8faff
channelId	712398d4-0cd9-4363-b53c-cde224816059
targetId	subscriberS1@broadsoft.com
EventData	Not applicable
call	Not applicable
callId	callhalf-62147:0
extTrackingId	446:1
personality	Terminator
state	Alerting
remoteParty	Not applicable
name	EMY -agentA1FirstName agentA1LastName
address	tel:1020
userId	agentA1@broadsoft.com
userDN	tel:+15006001020;ext=1020
callType	Group
diversionInhibited	Not applicable
startTime	1271789995718
agentEscalation	Emergency

F6 EventResponse

Parameter Name	Parameter Value
requestId	0:250
payload	Not applicable
EventResponse	Not applicable
eventID	2242376b-d19a-4ce1-bdff-64b85aab500b
statusCode	200
reason	OK

F7 HookStatusEvent

Parameter Name	Parameter Value
RequestId	0:251



Parameter Name	Parameter Value
Payload	Not applicable
Event	Not applicable
EventID	68ea76c2-660e-43b1-b1ff-f32931db2856
sequenceNumber	10
UserId	admin@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	573c743f-5561-4748-b46f-4d1fe1b8faff
channelId	712398d4-0cd9-4363-b53c-cde224816059
Targeted	subscriberS1@broadsoft.com
eventData	Not applicable
hookStatus	Off-Hook

F8 EventResponse

Parameter Name	Parameter Value
requestId	0:251
Payload	Not applicable
EventResponse	Not applicable
eventide	68ea76c2-660e-43b1-b1ff-f32931db2856
statusCode	200
reason	OK

F9 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:252
payload	Not applicable
Event	Not applicable
eventID	9aead070-efbd-4ff3-a538-c33cf9b75a23
sequenceNumber	9
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	573c743f-5561-4748-b46f-4d1fe1b8faff
channelId	712398d4-0cd9-4363-b53c-cde224816059
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-62147:0

Parameter Name	Parameter Value
extTrackingId	446:1
personality	Terminator
state	Active
remoteParty	Not applicable
name	EMY -agentA1FirstName agentA1LastName
address	tel:1020
userId	agentA1@broadsoft.com
userDN	tel:+15006001020;ext=1020
callType	Group
endpoint	Not applicable
addressOfRecord	5006001011@broadsoft.com
appearance	1
diversionInhibited	Not applicable
startTime	1271789995718
answerTime	1271789997446
agentEscalation	Emergency

F10 EventResponse

Parameter Name	Parameter Value
requestId	0:252
payload	Not applicable
EventResponse	Not applicable
eventID	9aead070-efbd-4ff3-a538-c33cf9b75a23
statusCode	200
reason	OK

F11 CallAnsweredEvent

Parameter Name	Parameter Value
requestId	0:253
payload	Not applicable
Event	Not applicable
eventID	b25acfce-789a-4fd3-b53a-0a17aab7bb57
sequenceNumber	31
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	f64ffa17-5341-43ad-81a3-a73113f170da

Parameter Name	Parameter Value
channelId	712398d4-0cd9-4363-b53c-cde224816059
targetId	agentA1@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-62131:1
extTrackingId	446:1
personality	Originator
state	Active
remoteParty	Not applicable
name	subscriberS1FirstName subscriberS1LastName
address	tel:1011
userId	subscriberS1@broadsoft.com
userDN	tel:+15006001011;ext=1011
callType	Group
diversionInhibited	Not applicable
startTime	1271789995718
answerTime	1271789997446
agentEscalation	Emergency

F12 EventResponse

Parameter Name	Parameter Value
requestId	0:253
payload	Not applicable
EventResponse	Not applicable
eventId	b25acfce-789a-4fd3-b53a-0a17aab7bb57
statusCode	200
reason	OK

F13 CallUpdatedEvent

Parameter Name	Parameter Value
requestId	0:254
payload	Not applicable
Event	Not applicable
eventId	677e6022-1aeb-4a34-ab32-38e43d0621ef
sequenceNumber	32
userId	adminCTI@broadsoft.com@broadsoft.com



Parameter Name	Parameter Value
externalApplicationId	AppCtlId
subscriptionId	f64ffa17-5341-43ad-81a3-a73113f170da
channelId	712398d4-0cd9-4363-b53c-cde224816059
targetId	agentA1@broadsoft.com
eventData	Not applicable
call	Not applicable
callId	callhalf-62131:0
extTrackingId	443:1
personality	Terminator
state	Active
remoteParty	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network
startTime	1271789958613
answerTime	1271789960056
acdCallInfo	Not applicable
acdUserId	ACD_CTL1@broadsoft.com
acdName	ACD_CTL1_Name
acdNumber	tel:+15006001025
numCallsInQueue	0
waitTime	0
callingPartyInfo	Not applicable
name	NetworkN1FirstName NetworkN1LastName
address	tel:+15007003333
callType	Network

F14 EventResponse

Parameter Name	Parameter Value
requestId	0:254
payload	Not applicable
EventResponse	Not applicable
eventID	677e6022-1aeb-4a34-ab32-38e43d0621ef
statusCode	200
reason	OK

F15 ConferenceStartedEvent

Parameter Name	Parameter Value
requestId	0:255
payload	Not applicable
Event	Not applicable
eventID	28825748-39cd-429a-8926-cb02bc7ca3bf
sequenceNumber	33
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	f64ffa17-5341-43ad-81a3-a73113f170da
channelId	712398d4-0cd9-4363-b53c-cde224816059
targetId	agentA1@broadsoft.com
eventData	Not applicable
conference	Not applicable
state	Active
endpoint	Not applicable
addressOfRecord	5006001020@broadsoft.com
appearance	1
conferenceParticipantList	Not applicable
conferenceParticipant	Not applicable
callId	callhalf-62131:0
conferenceParticipant	Not applicable
callId	callhalf-62131:1

F16 EventResponse

Parameter Name	Parameter Value
requestId	0:255
payload	Not applicable
EventResponse	Not applicable
eventID	28825748-39cd-429a-8926-cb02bc7ca3bf
statusCode	200
reason	OK

9.6 Service Management Message Flows

The following subsections illustrate the following scenarios:

- Call Forwarding Always (CFA) service change
- Do Not Disturb service change

- Hoteling Host and Guest service change
- Voice Mail Message Summary service change

9.6.1 Call Forwarding Always

This scenario illustrates a successful Call Forwarding Always provisioning for Subscriber S1.

Activity	Monitored User S1	Comments
The remote application sends a Modify Call Forwarding Always Request for subscriber S1.	Modify Call Forwarding Always Request (F1) Response (F4)	
A notification is sent for the new CFA configuration parameters.	Call Forwarding Always Event (F2) Response (F3)	

F1 Request

Parameter Name	Parameter Value
requestId	11
sessionId	MySessionId
credentials	YWRtaW5DVEIAjYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
GetCallForwardingAlwaysRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/services/CallForwardingAlways
method	PUT
version	Not applicable
params	Not applicable
subscriberId	subscriberS1@broadsoft.com
payload	Not applicable
CallForwardingAlways	Not applicable
active	true
forwardToPhoneNumber	1020
ringSplash	true

F2 CallForwardingAlwaysEvent

Parameter Name	Parameter Value
requestId	9:301
payload	Not applicable
Event	Not applicable
eventID	5eda485f-48e0-4a4e-9960-e51c8af02673
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com

Parameter Name	Parameter Value
externalApplicationId	AppCtlId
subscriptionId	8ffb5dee-d982-4b64-a40a-b4667d7b5eda
channelId	027f69b3-79f1-4671-8440-952e203a2d7d
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
info	Not applicable
active	true
forwardToPhoneNumber	tel:1020
ringSplash	true

F3 EventResponse

Parameter Name	Parameter Value
requestId	9:301
payload	Not applicable
EventResponse	Not applicable
eventID	5eda485f-48e0-4a4e-9960-e51c8af02673
statusCode	200
reason	OK

F4 Response

Parameter Name	Parameter Value
requestId	11
sessionId	116
statusCode	200
reason	OK
payload	Not applicable

9.6.2 Do Not Disturb

This scenario illustrates a successful Do Not Disturb provisioning for Subscriber S1.

Activity	Monitored User S1	Comments
The remote application sends a Modify Do Not Disturb Request for subscriber S1.	Modify Do Not Disturb Request (F1) Response (F4)	
A notification is sent for the new Do Not Disturb configuration parameters.	Do Not Disturb Event (F2) Response (F3)	



F1 Request

Parameter Name	Parameter Value
requestId	13
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
ModifyDoNotDisturb Request	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/services/donotdisturb
method	PUT
version	Not applicable
params	Not applicable
subscriberId	subscriberS1@broadsoft.com
payload	Not applicable
DoNotDisturb	Not applicable
active	true
ringSplash	true

F2 DoNotDisturbEvent

Parameter Name	Parameter Value
requestId	9:316
payload	Not applicable
Event	Not applicable
eventID	d2031640-bd7e-4131-9974-7fa529aea361
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	680d3762-75ba-4984-a60c-38ea1ee6cff0
channelId	027f69b3-79f1-4671-8440-952e203a2d7d
targetId	subscriberS1@broadsoft.com
EventData	Not applicable
info	Not applicable
active	true
ringSplash	true

F3 EventResponse

Parameter Name	Parameter Value
requestId	9:316
payload	Not applicable

Parameter Name	Parameter Value
EventResponse	Not applicable
eventID	d2031640-bd7e-4131-9974-7fa529aea361
statusCode	200
reason	OK

F4 Response

Parameter Name	Parameter Value
requestId	13
sessionId	128
statusCode	200
reason	OK
payload	Not applicable

9.6.3 Hoteling Host and Guest (AS only)

This scenario illustrates a successful association of a Hoteling Guest with a Hoteling Host. After the association, the guest user (subscriber S1) can be reached at the host's device.

Activity	Monitored User S1	Monitored User S2	Comments
The remote application fetches the available hosts for subscriber S1.	Get Hoteling Host List Request (F1) Response (F2)		The application obtains a list of potential hosts.
The remote application sends a Modify Hoteling Guest Request for subscriber S1.	Modify Hoteling Guest Request (F3) Response (F4)		SubscriberS2 is selected as the target host.
A notification is sent for the new Hoteling Guest parameters.	Hoteling Guest Request (F5) Response (F6)		

F1 Request

Parameter Name	Parameter Value
requestId	15
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
GetHotelngHosts Request	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/directories/hotelnghosts
method	GET
version	Not applicable
params	Not applicable
subscriberId	subscriberS1@broadsoft.com



Parameter Name	Parameter Value
payload	Not applicable

F2 Response

Parameter Name	Parameter Value
requestId	15
sessionId	134
statusCode	200
reason	OK
payload	Not applicable
AvailableHotelingHosts	Not applicable
hostUserIdList	Not applicable
userId	subscriberS2@broadsoft.com

F3 Request

Parameter Name	Parameter Value
requestId	16
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
ModifyHotelingGuestRequest	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/services/hotelinguuest
method	PUT
version	Not applicable
params	Not applicable
subscriberId	subscriberS1@broadsoft.com
payload	Not applicable
HotelingGuest	Not applicable
active	true
enableAssociationLimit	true
associationLimitHours	12
hostUserId	subscriberS2@broadsoft.com

F4 Response

Parameter Name	Parameter Value
requestId	16
sessionId	135



Parameter Name	Parameter Value
statusCode	200
reason	OK
payload	Not applicable

F5 HotelingGuestEvent

Parameter Name	Parameter Value
requestId	9:338
payload	Not applicable
Event	Not applicable
eventID	896fff0a-1113-434c-b665-fb9c036482ed
sequenceNumber	2
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	ba7fc9e8-85f7-436e-973b-3b2a8bd16561
channelId	027f69b3-79f1-4671-8440-952e203a2d7d
targetId	subscriberS1@broadsoft.com
eventData	Not applicable
info	Not applicable
active	true
enableAssociationLimit	true
associationLimitHours	12
hostUserId	subscriberS2@broadsoft.com

F6 EventResponse

Parameter Name	Parameter Value
requestId	9:338
payload	Not applicable
EventResponse	Not applicable
eventID	896fff0a-1113-434c-b665-fb9c036482ed
statusCode	200
reason	OK

9.6.4 Voice Mail Message Summary (AS only)

This scenario illustrates a successful Clear Voice Mail Message Summary command for Subscriber S1.

Before clearing Message Summary	After clearing Message Summary
S1 newMessages > 0	S1 newMessages = 0

Figure 158 Clear Voice Mail Message Summary

Activity	Monitored User S1	Comments
The remote application sends a Clear Voice Mail Message Summary Request for subscriber S1.	Clear Voice Mail Message Summary Request (F1) Response (F2)	
A Message Summary notification is sent for subscriber S1.	Voice Mail Message Summary Event (F3) Response (F4)	The number of new messages has been reset.

F1 Request

Parameter Name	Parameter Value
requestId	23
sessionId	MySessionId
credentials	YWRtaW5DVEIAYnJvYWRzb2Z0LmNvbTpwYXNzd29yZA==
ClearVoiceMail MessageSummary Request	Not applicable
uri	/com.broadsoft.xsi-actions/v2.0/user/%subscriberId%/calls/messaging/VoiceMailMessageSummary
method	DELETE
version	Not applicable
params	Not applicable
subscriberId	subscriberS1@broadsoft.com
payload	Not applicable

F2 Response

Parameter Name	Parameter Value
requestId	23
sessionId	224
statusCode	200
reason	OK
payload	Not applicable



F3 VoiceMailMessageSummaryEvent

Parameter Name	Parameter Value
requestId	21:466
payload	Not applicable
Event	Not applicable
eventID	cd0dc6cd-45f3-472c-8a26-114557898916
sequenceNumber	3
userId	adminCTI@broadsoft.com@broadsoft.com
externalApplicationId	AppCtlId
subscriptionId	ea8a5fa3-8d8e-4469-ae1a-54c0da593086
channelId	04074f9c-b8f6-4431-9f80-4228a674a9b5
targetId	subscriberS1@broadsoft.com
EventData	Not applicable
messageSummary	Not applicable

F4 EventResponse

Parameter Name	Parameter Value
requestId	21:466
payload	Not applicable
EventResponse	Not applicable
eventID	cd0dc6cd-45f3-472c-8a26-114557898916
statusCode	200
reason	OK

10 Error Code List

The following table lists, in ascending order, all *error codes* that can be returned by Cisco BroadWorks.

errorCode	Summary
1000	Missing required parameter.
1001	Invalid response from server.
1100	Invalid value of <i>startIndex</i> or number of results. Minimum value of <i>startIndex</i> and number of results should be 1.
1101	Non-HTTPs request.
1102	Missing entity body in request.
1103	Failed to parse XPath information.
1104	Unexpected value for format.
1105	Callback not acceptable.
1106	Callback expected.
1107	Internal server error, session ended before request processing.
1108	Method not supported.
1109	Unable to read from request.
1110	User name missing in URL.
1111	Internal server error – unexpected error while reading from request.
1112	Parsing xml-fragment body fails.
1113	Service handler not found.
1114	Unable to read JSON from request.
1115	Unexpected exception while providing JSON response.
1116	Invalid Xml.
1123	Comet request required to create a channel.
100001	User with address <address> is not found.
100002	Invalid user ID <userId>. The user ID must be of the form user@domain.
100003	User <userId> not found.
100004	Unauthorized request.
100005	Address <address> must be a SIP or tel URI.
100006	Endpoint state is not valid.
100007	Blocked by Outgoing Calling Plan service.
100008	Remote party for call <callId> is a virtual subscriber.
100009	Blocked by Communication Barring service.
100010	Blocked by translations.
100011	Blocked by the user's Privacy service.



errorCode	Summary
100012	User for address <address> has no primary phone number.
100013	Address <address> cannot be an international E.164 address.
100014	Admin/User <userId> is not authorized for user <userId>.
100015	User is a virtual subscriber.
100016	Blocked by Account/Authorization Codes service.
100017	Blocked by Maximum Concurrent Redirections policy.
100018	Blocked by Intercept Group or Intercept User service.
100019	Blocked by Maximum Simultaneous Calls policy.
100020	No endpoint for call <callId>.
100021	Endpoint state for call <callId> is not valid.
100022	Blocked by Zone Calling Restrictions.
100023	Blocked by Incoming Calling Plan service.
100024	Blocked by Hierarchical Communication Barring service.
100026	Blocked by Session Admission Control group.
101000	Call <callId> has no remote call.
101001	State for call <callId> is not valid.
101002	Call <callId> not found.
101003	Call <callId> is an Emergency call.
101004	Call <callId> is in the detached state.
101005	Diversions are inhibited for call {0}.
101006	Call <callId> is not valid.
101007	Call <callId> does have the expected session ID of <sessionId>.
101008	Invalid call IDs.
102000	System error.
102001	Denied due to overload.
102002	Unable to find or create session.
102003	XML parsing failed: <message>.
102004	Invalid encoding: <encoding>.
102005	XML schema validation failed: <errorList>.
102006	Request is not valid.
102007	Request handler not found.
102008	Application ID <applicationId> is unknown.
102009	Host <host> is not in access control list (ACL) for application ID <applicationId>.
102010	Application id <applicationId> does not match currently registered application ID.
110000	Transfer target is not valid for the transfer to voice mail command.



errorCode	Summary
110001	Error communicating with Voice Messaging repository.
110002	Message download is already in progress.
110003	Message <messageId> not found.
110010	Answer confirmation in progress for call <callId>.
110011	Answer confirmation in progress.
110030	User is not assigned the Automatic Callback service.
110040	User is not assigned the BroadWorks Anywhere service.
110041	BroadWorks Anywhere service is not activated for this user.
110060	Customer Originated Trace already issued for call <callId>.
110061	No valid call found to perform Customer Originated Trace.
110062	User is not assigned the Customer Originated Trace service.
110070	User is not assigned the Call Forwarding Always service.
110080	User is not assigned the Call Forwarding Busy service.
110090	User is not assigned the Call Forwarding No Answer service.
110100	Cannot apply a disposition code to call {0} since it is not a call center call.
110101	Invalid DNIS Address.
110102	Inactive Disposition Code.
110103	User <userId> is not a supervisor for this agent.
110104	All supervisors for this agent are busy or cannot receive an emergency call.
110105	User is not a supervisor.
110106	Agent must be in <i>Wrap-up</i> state when call is not specified.
110107	Disposition code is not valid for this call center.
110108	Agent is not assigned the Call Center-Premium service.
110109	Disposition codes disabled for this call center.
110110	Emergency escalation in progress.
110200	User is not assigned the Call Center-Basic, the Call Center-Standard, or the Call Center-Premium service.
110201	User is not assigned the Call Center-Standard or the Call Center-Premium service.
110202	User is not assigned the Call Center-Premium service.
110203	User is not a call center.
110204	User is not an agent or supervisor of that call center.
110205	Cannot perform function - call is bounced.
110206	Cannot perform function - mandatory entrance message is playing.
110207	Cannot perform function - only calls in the highest priority can be reordered.
110208	Transfer target must be a Call Center when using Transfer to front function.



errorCode	Summary
110209	Only calls in a premium call center can be promoted.
110210	Calls can only be promoted to a higher priority.
110400	Agent <userId> not found.
110401	An outgoing call for a route point cannot be failed.
110402	Route Point is failed.
110403	Failover policy is disabled.
110404	Route point is inactive.
110405	Route point name <RPName> is not valid.
110406	Route point queue is full.
110407	User is not a route point.
110408	Per call failure policy is disabled.
110409	Play collect ID <playCollectId> not found.
110410	Failover in progress
110411	Offered call is configured to play ringback and cannot have a treatment applied.
110412	External system is not ready.
110500	User is not assigned the Call Transfer service.
110501	Consultation transfer in progress.
110502	Transfer has failed.
110503	Consultative transfer already in progress.
110504	Remote party for call <callId> cannot be transferred while the transfer target is Alerting.
110505	Calls are not in proper state. Call states cannot be Alerting for both calls.
110506	Transfers are inhibited for call <callId>.
110507	Call <callId> with recall type <recallType> cannot be redirected prior to answer.
110520	User's group is not assigned the Call Park service.
110521	Retrieving user has no primary extension or DN.
110522	Call released during park attempt.
110523	Call is not in a parkable condition.
110524	Could not find a call in parkable condition.
110525	User is not a member of a Group Call Park group.
110526	Parking user has no primary extension or DN.
110527	Remote call rejected the park attempt.
110528	Call to park no longer exists.
110529	Failed to collect digits.
110530	Park against address is not valid.
110531	Cannot park a call against a Virtual Subscriber.



errorCode	Summary
110532	Remote party does not support Call Park.
110533	Park against user is the remote party.
110534	Call already parked against user.
110535	Blocked by Zone Calling Restrictions - cannot park Network call against inter-zone target.
110535	No available Group Call Park member.
110540	The call <callId> is a Push-To-Talk call.
110541	Call <callId> is an answered Push-To-Talk call.
110542	User is not assigned the Push-to-Talk service.
110550	User is not a member of a Call Pickup Group.
110551	User is not assigned the Directed Call Pickup service.
110552	User is not assigned the Directed Call Pickup with Barge-in service.
110560	User is not assigned the Call Return service.
110570	User is not assigned the Call Waiting service.
110580	User has no charge address.
110581	Emergency call is present.
110582	Blocked by prepaid service.
110610	Call <callId> is part of a conference.
110620	Call <callId> is alerting but answer is not supported by the device or endpoint.
110630	User is not assigned the CommPilot Express service.
110640	No conference present.
110641	Conference release in progress.
110642	Conference already present.
110643	Number of participants <numberParticipants> exceeds maximum of <maxValue>.
110644	Calls are not in proper state. Calls must be in the <i>Alerting</i> , <i>Active</i> , <i>Held</i> , or <i>Remote Held</i> state with personality Originator or Terminator. There can only be one Alerting call with Terminator personality.
110645	Call <callId> is not in the conference.
110646	Not allowed to have more than one call outside the conference.
110647	Conference is already held.
110648	Call <callId> is already in the conference.
110649	Already at max participants of <numberParticipants>.
110650	User is not assigned the Three-Way Call nor N-Way Call service.
110651	Conference is already active.
110690	User is not assigned the Do Not Disturb service.
110710	User is not assigned the Hoteling Guest service.



errorCode	Summary
110720	User is not assigned the Last Number Redial service.
110730	User is not assigned the Music On Hold User service.
110740	User is not assigned the N-Way Call service.
110750	User is not assigned the Sequential Ring service.
110760	User is not assigned the Call Center Monitoring service.
110761	Monitor next call already in progress for the target.
110770	User is not assigned the Simultaneous Ring Personal service.
110780	User is not assigned the Third-Party Voice Mail Support service.
110790	User is not assigned the Three-Way Call service.
110800	User is not assigned the Voice Messaging User service nor Third-Party Voice Mail Support service or the service is not enabled.
110801	User is not assigned the Voice Messaging User service.
110802	User is not assigned the Voice Messaging User nor Third-Party Voice Mail Support service.
110810	DTMF transmission already in progress.
110811	DTMF string <dtmfString> is not valid. The string can contain only the following characters: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, #, A, B, C, D ;'.
110820	User is not assigned the Remote Office service.
110821	Blocked by Remote Office.
110830	Application controller <AppControllerId> not found.
110831	No channels for application controller <AppControllerId>.
110841	Channel <channelId> not found.
110842	Channel set <channelSetId> not found.
110860	Group <groupId> not found.
110861	Service provider or enterprise <serviceProviderId or enterpriseId> not found <dtmfString>.
110862	Subscription ID <subscriptionId> not found.
110863	HttpContact or ChannelSetId must be defined.
110864	Event package is deprecated for the release specified.
110870	Profile Server address is either inactive or invalid.
110871	This conference is not active; Outdial failed.
110872	Conference has expired.
110873	Invalid Conference ID.
110874	Another moderator level operation is in progress. Try later.
110875	Participant has declined the outdial request.
110876	Outdial request with invalid address.
110877	Participant does not exist.



errorCode	Summary
110878	Participant on hold cannot be muted or unmuted.
110882	Recording is not enabled for this conference.
110883	This conference is already being recorded.
110884	This conference is not in paused state.
110885	This conference is paused; it can either resume or stop.
110886	This conference is not being recorded.
110888	Target User is not a Meet-Me virtual subscriber.
110889	Conference has not started.
110890	The number of unmuted participants has reached the maximum limit. You cannot unmute now.
110891	Recording cannot be started until the conference has started.
110892	The number of conference attendees has reached the maximum limit. New attendees cannot join the conference.
110893	The server hosting the conference is not available.
110970	Target user does not have the Call Me Now service or the service is disabled.
110971	Call Me Now calls may not be placed to an emergency or repair address.
110972	Call Me Now calls may not be to an address of the target user.
110973	Call Me Now call has been rejected by screening.
110980	User is not assigned the Call Recording service.
110981	User recording mode is never.
110982	Recording failed; experiencing problems connecting to the call recording platform.
110983	User recording mode does not support <command>.
110984	Recording is not active.
110985	Outstanding request in progress.
111000	Invalid location value <location>.
111001	Cannot find location <location>.
111002	locationAddress not allowed for location <location>.
111003	User is not assigned the <serviceName> service or there is no location assigned to this service.
111004	User does not have a primary endpoint.
111010	Target user does not have a multiple appearance service enabled.
111011	No call to retrieve.
111012	Call already assigned to location.
111020	Neither the BroadWorks Mobility service nor the BroadWorks Anywhere service is assigned to this user.
111021	Neither the BroadWorks Mobility service nor the BroadWorks Anywhere service is activated for this user.



errorCode	Summary
111022	CallingPartyAddress <address> not found.
111023	Duplicate request for same call context.
111024	IMRN request not allowed.
111025	IMRN allocation failed.
111030	User is not a member of Find-Me/Follow-Me group.
111031	Not a Find-Me/Follow-Me call.
111110	Executive Assistant Service not assigned to this user.
111111	Non-executive call for assistant.
111112	This call is not answered by the assistant.
111113	Executive user <userId> not found.
111114	Assistant cannot transfer the executive call back to the executive.
111040	Security Classification service is not assigned to this user.
111041	Classification level is not allowed for the user.
111042	Security Classification override is not allowed because the user is not on a call.
111043	The conference does not have a security classification.
111120	User is not assigned the Flexible Seating Guest service.
111200	The user does not have the Collaborate – Audio service assigned.
111201	The room <roomid> does not exist.
111202	Cannot delete room of type My Room. Only rooms of type Instant Room and Project Room can be deleted.
111203	The room is locked. Ask the owner to unlock the room.
111204	A room of type Instant Room cannot reset its roomID.
111205	Invalid room type requested.
111206	Collaborate Room Not Started.
111207	Collaborate Room Hosting Server not available.
111208	Collaborate Room has Expired.
111209	Collaborate Room Participant declined outdial request.
111210	Invalid Outdial Address.
111211	The participant does not exist.
111212	Room has maximum number of participants. Unable to add participant.
111250	User is not assigned the 3G/4G Continuity service.
111251	3G/4G Continuity service is not activated for this user.

11 Acronyms and Abbreviations

ACD	Automatic Call Distribution
ACL	Access Control List
AS	Application Server
BOM	Byte Order Marker
CDR	Call Detail Record
CFA	Call Forwarding Always
CFB	Call Forwarding Busy
CFNA	Call Forwarding No Answer
CFNR	Call Forwarding Not Reachable
CFS	Call Forwarding Selective
CLID	Calling Line Identification
CLIP	Calling Line Identification Presentation
COLP	Connected Line Identification Presentation
COT	Customer Originated Trace
CPD	Call Progress Detection
CPE	Customer Premises Equipment
CTI	Computer Telephony Integration
DN	Directory Number
DND	Do Not Disturb
DNIS	Dialed Number Identification Service
DPU	Directed Pickup
DPUBI	Directed Call Pickup with Barge-in
DTMF	Dual-Tone Multi-Frequency
FAC	Feature Access Code
FIFO	First in First out
FMFM	Find-me/Follow-me
IMRN	IP Multimedia Routing Number
IMS	IP Multimedia Subsystem
IP	Internet Protocol
ISDN	Integrated Services Digital Network
JSON	JavaScript Object Notation
MGCP	Media Gateway Control Protocol
MOH	Music On Hold
MWI	Message Waiting Indicator



SIP	Session Initiation Protocol
SIT	Special Information Tone
SMS	Short Message Service
SNMP	Simple Network Management Protocol
SSID	Shared Subscription ID
TAS	Telephony Application Server
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UCaaS	Unified Communications as a Service
URI	Uniform Resource Identifier
VM	Voicemail
VoIP	Voice over Internet Protocol
XS	Execution Server
Xsp	Xtended Services Platform



12 References

- [1] Cisco Systems, Inc. 2019. *Cisco BroadWorks Treatment Guide, Release 23.0*. Available from Cisco at xchange.broadsoft.com.
- [2] W3C Recommendation 28 October 2004. *XML Schema Part 2: Data Types Second Edition*. Available at <http://www.w3.org/TR/xmlschema-2/>.
- [3] Cisco Systems, Inc. 2019. *Cisco BroadWorks Service Guide, Release 23.0*. Available from Cisco at xchange.broadsoft.com.
- [4] Cisco Systems, Inc. 2019. *Cisco BroadWorks Shared Call Appearance Interface Specification, Release 23.0*. Available from Cisco at xchange.broadsoft.com.
- [5] Cisco Systems, Inc. 2019. *Cisco BroadWorks Application Server Group Web Interface Administration Guide – Part 2, Release 23.0*. Available from Cisco at xchange.broadsoft.com.
- [6] F. Andreasen, B. Foster, *Media Gateway Control Protocol (MGCP) Version 1.0, Request for Comments: 3435*, IETF Network Working Group, January 2003 <http://www.rfc-editor.org/rfc/rfc3435.txt>.
- [7] Cisco Systems, Inc. 2019. *Cisco BroadWorks Hosted Thin Call Center Agent/Supervisor User Guide, Release 23.0*. Available from Cisco at xchange.broadsoft.com.
- [8] Cisco Systems, Inc. 2019. *Cisco BroadWorks SIP Access Side Extensions Interface, Release 23.0*. Available from Cisco at xchange.broadsoft.com.
- [9] Cisco Systems, Inc. 2019. *Cisco BroadWorks Accounting CDR Interface Specification, Release 23.0*. Available from Cisco at xchange.broadsoft.com.
- [10] Johnston, A., Levin, O., “*Session Initiation Protocol (SIP) Call Control – Conferencing for User Agents*”, Request for Comments 4579, Internet Engineering Task Force, August 2006. Available from <http://www.ietf.org/>.
- [11] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., “*HyperText Transfer Protocol—HTTP/1.1*”, RFC 2616, Internet Engineering Task Force, June 1999. Available from <http://www.ietf.org/>.
- [12] Cisco Systems, Inc. 2019. *Cisco BroadWorks Feature Overview, Release 23.0*. Available from Cisco at xchange.broadsoft.com.
- [13] Cisco Systems, Inc. 2019. *Cisco BroadWorks XS Mode Configuration Guide, Release 23.0*. Available from Cisco at xchange.broadsoft.com.
- [14] Cisco Systems, Inc. 2019. *Cisco BroadWorks Mobility Guide, Release 23.0*. Available from Cisco at xchange.broadsoft.com.
- [15] Cisco Systems, Inc. 2018. *Simplify Notifications to UCaaS and Collaborate Feature Description, Release 23.0*. Available from Cisco at xchange.broadsoft.com.