



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Voice Microservices Events and Models Reference

Table of Contents

Topics	
Voice Microservices topics for event streaming	6
Events	
Agent state and DN events	9
Call treatment events	22
Call routing events	28
Call handling and transfer/conference events	33
Event attributes	
Event attributes	54
Call models and flows	
Basic call models	75
Releasing calls	90
Holding, transferring, and conferencing	93
Predictive dialing	110
Monitoring calls	119
Working with queues	125
User data	
Handling user data	132

Contents

- [1 Topics](#)
- [2 Events](#)
- [3 Call models](#)

Find links to all the topics in this guide.

Related documentation:

-
-

RSS:

- [For private edition](#)

This document introduces you to the agent- and call-related events and models that you might encounter in a Genesys Voice Microservices deployment. You will find the following information in this guide:

- A list of call events and their descriptions.
- A list of agent state and DN events and their descriptions.
- A collection of common call models and flows.

Topics

Learn about the topics that Voice Microservices use for event streaming.

- [Topics](#)

Events

Learn about the events produced by Voice Microservices.

- Agent state and DN events
 - Call treatment events
 - Call routing events
 - Call handling and transfer/conference events
 - Event attributes
-

Call models

Learn about the Voice Microservices call models.

- Basic call models
 - Releasing calls
 - Holding, transferring, and conferencing
 - Predictive dialing
 - Monitoring calls
 - Working with queues
-

Voice Microservices topics for event streaming

Contents

- 1 Events stored in each topic
 - 1.1 Events for the voice-agentstate topic
 - 1.2 Events for the voice-callthread topic

Learn about the topics that Voice Microservices use for event streaming.

Related documentation:

-
-
-

RSS:

- [For private edition](#)

Voice Microservices use the following topics for event streaming:

- voice-agentstate
 - To consume agent-related events, a client's consumer must subscribe to the voice-agentstate topic.
 - The event key uses the "{tenant_id}:{agent_id}" template.
- voice-callthread
 - To consume call-related events, a client's consumer must subscribe to the voice-callthread topic. Genesys recommends that you avoid using the **callthread** consumer group name for subscription.
 - The event key uses the "{tenant_id}:{callthread_id}" template.

Events stored in each topic

This section lists the events for each Voice Microservices topic.

Voice Microservices events are in JSON format. This is a sample of the EventAgentLogin event and its attributes:

```
{ "id": EventAgentLogin, AgentID, ThisDN, AgentWorkMode, Extensions: {} }
```

The word "group", where used in this documentation, does not refer to any part of the topic or event structure. The group names are used to collect together similar events based on the purpose of events and the type of information they convey.

Events for the voice-agentstate topic

For information about the events in the voice-agentstate topic, including event descriptions and specific event attributes, see Agent state and DN events.

This is the list of events in the **Agent state and DN** group:

- EventAgentLogin
- EventAgentLogout
- EventAgentNotReady
- EventAgentReady
- EventDNBackInService
- EventDNDOff
- EventDNDOOn
- EventDNOutOfService
- EventForwardCancel
- EventForwardSet
- EventMuteOff
- EventMuteOn
- EventOffHook
- EventOnHook

Events for the voice-callthread topic

For information about the events in the voice-callthread topic, including event descriptions and specific event attributes, see the following pages:

- Call handling and transfer/conference events
- Call routing events
- Call treatment events

This is the list of events in the **Call handling and transfer/conference** group:

- EventAbandoned
- EventDestinationBusy
- EventDialing
- EventDiverted
- EventEstablished
- EventHeld
- EventNetworkReached
- EventPartyAdded
- EventPartyChanged
- EventPartyDeleted
- EventQueued
- EventReleased
- EventRetrieved
- EventRinging

This is the list of events in the **Call routing** group:

- EventRouteRequest
- EventRouteUsed

This is the list of events in the **Call treatment** group:

- EventTreatmentApplied
- EventTreatmentEnd
- EventTreatmentNotApplied

Agent state and DN events

Contents

- [1 EventAgentLogin](#)
- [2 EventAgentLogout](#)
- [3 EventAgentReady](#)
- [4 EventAgentNotReady](#)
- [5 EventDNOutOfService](#)
- [6 EventDNBackInService](#)
- [7 EventDNDOn](#)
- [8 EventDNDOff](#)
- [9 EventForwardSet](#)
- [10 EventForwardCancel](#)
- [11 EventOffHook](#)
- [12 EventOnHook](#)
- [13 EventMuteOn](#)
- [14 EventMuteOff](#)

Find information about the agent state and DN events that Voice Microservices produce.

Related documentation:

-
-
-

RSS:

- [For private edition](#)

Agent state and DN events are stored in the **voice-agentstate** topic.

Voice Microservices support the following agent state and DN events, which are described on this page. Click the attribute names in the tables below to see fuller descriptions of the attributes.

- | | | |
|--|---------------------------------------|--------------------------------|
| • EventAgentLogin | • EventDNDOff | • EventMuteOff |
| • EventAgentLogout | • EventDNDOOn | • EventMuteOn |
| • EventAgentNotReady | • EventDNOutOfService | • EventOffHook |
| • EventAgentReady | • EventForwardCancel | • EventOnHook |
| • EventDNBackInService | • EventForwardSet | |

EventAgentLogin

The agent has logged in to the ACD group specified by **ThisQueue**. Multiple agent logins are allowed for the same DN and agent ID combination (since EventAgentLogin does not indicate by itself a transition of agent state).

AgentID must be present if the agent is logged in through Voice Microservices or if the information is available.

If present, the **Extensions** attribute might include a ReasonCode value specifically used to communicate hardware reasons.

Event attribute	Short description
Event attribute	Short description

"> id (mandatory) The event name."> Server (mandatory) A unique identifier

assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> Agent ID (optional) This parameter uniquely identifies the ACD agent."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices."> ThisQueue (optional) The directory number of the most significant ACD group with respect to the event in question.

[>> Back to list](#)

EventAgentLogout

The agent has logged out of the ACD group specified by **ThisQueue**.

On CTI platforms that support agent login for multiple queues, this event signals that the agent has been moved to the Logged Out state, and is therefore used only for an agent's final logout.e

AgentID must be present if the agent is logged in through Voice Microservices or if the information is available.

If present, the **Extensions** attribute might include a ReasonCode value specifically used to communicate hardware reasons.

Event attribute	Short description
Event attribute	Short description

"> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to

the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> Agent ID (optional) This parameter uniquely identifies the ACD agent."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices."> ThisQueue (optional) The directory number of the most significant ACD group with respect to the event in question.

[>> Back to list](#)

EventAgentReady

The agent is ready to receive ACD calls.

AgentID must be present if the agent is logged in through Voice Microservices or if the information is available.

If present, the **Extensions** attribute might include a ReasonCode value specifically used to communicate hardware reasons.

Event attribute	Short description
Event attribute	Short description

"> AgentWorkMode (mandatory) This attribute indicates the agent/supervisor-related current work mode."> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> Extensions

(optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices."> ThisQueue (optional) The directory number of the most significant ACD group with respect to the event in question.

[>> Back to list](#)

EventAgentNotReady

The agent is not ready to receive ACD calls.

AgentID must be present if the agent is logged in through Voice Microservices or if the information is available.

If present, the **Extensions** attribute might include a ReasonCode value specifically used to communicate hardware reasons.

Event attribute	Short description
Event attribute	Short description

"> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> AgentWorkMode (optional) This attribute indicates the agent/supervisor-related current work mode."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices

or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices."> ThisQueue (optional) The directory number of the most significant ACD group with respect to the event in question.

[>> Back to list](#)

EventDNOutOfService

The DN specified in the **ThisDN** attribute is out of service and cannot make or receive calls. This event is generated when an out-of-service state is first detected or when a new client registers on a DN known to be out of service.

When a DN is out of service, only the following Voice Microservices requests can be issued for it: client registration and unregistration, queries, agent login, and private service requests.

Voice Microservices return a TERR_OUT_OF_SERVICE error if called on to attempt a supported operation that cannot progress on an out-of-service DN.

When a DN goes out of service, Voice Microservices notify the user about the termination of active calls or change an agent state (not ready/logout) using normal events. The other applications should rely only on those events to change the DN/agent state.

Event attribute	Short description
Event attribute	Short description

"> id (mandatory) The event name."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request.

[>> Back to list](#)

EventDNBackInService

The DN specified in the **ThisDN** attribute is back in service and can make or receive calls. This event is generated when a DN, which has been out of service and for which the EventDNOutOfService was previously distributed, returns to service.

In the absence of EventDNOutOfService and EventDNBackInService, all clients should assume, for backward-compatibility reasons, that the DN is in service.

Between EventDNOutOfService and EventDNBackInService, the client is not able to perform any requests, and no events should be expected during this outage. Genesys recommends that you perform TQueryAddress() after EventDNBackInService to ensure synchronization between Voice Microservices and the client.

Event attribute	Short description
Event attribute	Short description

"> id (mandatory) The event name."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request.

[>> Back to list](#)

EventDNDoNotDisturb

The Do-Not-Disturb (DND) feature has been turned on for the telephony object specified by **ThisDN**.

Event attribute	Short description
Event attribute	Short description

"> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the

request a client sends to Voice Microservices.

[>> Back to list](#)

EventDNDOff

The Do-Not-Disturb (DND) feature has been turned off for the telephony object specified by **ThisDN**.

Event attribute	Short description
Event attribute	Short description

"> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices.

[>> Back to list](#)

EventForwardSet

The Forwarding feature has been turned on for the telephony object specified by **ThisDN**.

The **InfoStatus** event attribute has a value of either CallForwardingStatus or

SendAllCallsStatus in the EventForwardSet event.

The **OtherDN** event attribute specifies the target party when the Forward feature is in progress.

Event attribute	Short description
Event attribute	Short description

"> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> InfoStatus (optional) The InfoType information about the telephony object specified by **ThisDN** and/or **ThisQueue**."> OtherDN (optional) The directory number of the second most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices.

[>> Back to list](#)

EventForwardCancel

The Forwarding feature has been turned off for the telephony object specified by **ThisDN**.

Event attribute	Short description
Event attribute	Short description

"> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDN (mandatory) The directory number of the most

significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices.

[>> Back to list](#)

EventOffHook

The telephony object specified by **ThisDN** has gone off-hook.

Event attribute	Short description
Event attribute	Short description

"> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> CallHistory (optional) Information about transferring/routing of the call through a multisite contact center network."> CallID (optional) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> CallState (optional) The current status of the call the event relates to."> CallType (optional) The type of call in question."> ConnID (optional) A current connection identifier of the call to which this event relates."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that

cannot be described by the other parameters in an event or a request."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> ThisTrunk (optional) The identifier of the most significant trunk with respect to the event in question."> UserData (optional) Specifies the pointer to the call-related user data.

[>> Back to list](#)

EventOnHook

The telephony object specified by **ThisDN** has gone on-hook.

Event attribute	Short description
Event attribute	Short description

"> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> CallHistory (optional) Information about transferring/routing of the call through a multisite contact center network."> CallID (optional) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> ConnID (optional) A current connection identifier of the call to which this event relates."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> ThisDNRole (optional) The role of the telephony object specified by **ThisDN** in the event in question."> ThisTrunk (optional) The identifier of the most significant trunk with respect to the event in question."> UserData (optional) Specifies the pointer to the call-related user

data.

[>> Back to list](#)

EventMuteOn

A party identified by **ThisDN** is now in the Mute mode.

Event attribute	Short description
Event attribute	Short description

"> ConnID (mandatory) A current connection identifier of the call to which this event relates."> id (mandatory) The event name."> ThisDNRole (mandatory) The role of the telephony object specified by **ThisDN** in the event in question."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices."> UserData (optional) Specifies the pointer to the call-related user data.

[>> Back to list](#)

EventMuteOff

A party identified by **ThisDN** is no longer in Mute (microphone-disabled) mode. The **ReferenceID** attribute is set to indicate the corresponding TSetMuteOff() function.

Event attribute	Short description
Event attribute	Short description

"> ConnID (mandatory) A current connection identifier of the call to which this event relates."> id (mandatory) The event name."> ThisDNRole (mandatory) The role of the telephony object specified by **ThisDN** in the event in question."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices."> UserData (optional) Specifies the pointer to the call-related user data.

[>> Back to list](#)

Call treatment events

Contents

- [1 EventTreatmentApplied](#)
- [2 EventTreatmentEnd](#)
- [3 EventTreatmentNotApplied](#)

Find information about the call treatment events that Voice Microservices produce.

Related documentation:

-
-
-

RSS:

- [For private edition](#)

Call treatment events are stored in the voice-callthread topic.

Voice Microservices support the following call treatment events, which are described on this page. Click the attribute names in the tables below to see fuller descriptions of the attributes.

- [EventTreatmentApplied](#)
- [EventTreatmentEnd](#)
- [EventTreatmentNotApplied](#)

EventTreatmentApplied

The call has been treated and the Treatment Device (TD) is processing the treatment instruction.

Event attribute	Short description
Event attribute	Short description

"> CallID (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> CallThreadID (mandatory) The call thread identifier of the call."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call to which this event relates."> CustomerID (mandatory) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> Extensions (mandatory) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDNRole (mandatory) The role of the telephony object specified by **ThisDN** in the event in question."> ThisDN (mandatory) The

directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour).">

TreatmentType (mandatory) The type of treatment to be applied to the telephony object in question."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived.">

NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN.">

ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices."> TransferConnID (optional) Contains the **ConnID** of the call for which transfer or conference was initiated."> TreatmentParms (optional) Contains parameters to be used for the treatment.">

UserData (optional) Specifies the pointer to the call-related user data.

[>> Back to list](#)

EventTreatmentEnd

The call has been treated and the Treatment Device (TD) is waiting for another instruction.

This event does not appear in cases of continuing treatments like Silence or RingBack.

The **CollectedDigits** and **LastCollectedDigit** attributes are present if **TreatmentType** is either CollectDigits or PlayAnnouncementAndCollectDigits.

The following key-value pairs are set for all treatment types:

- For all treatment types where an announcement was played, INTERRUPTED is set to:
 - NO, if the announcement was not interrupted.
 - KEYPAD, if it was interrupted by keypad entry.
 - VOICE, if it was interrupted by the caller speaking something.
- For all treatment types where digits are to be collected from the caller, COMPLETION_STATUS is set to:
 - NORMAL, if the treatment completed normally (optional).
 - TIMEOUT, if the digit collection timed out before all required digits could be collected.
 - CANCELLED, if the treatment was cancelled by a request from router.

- For **TreatmentType**=DigitsVerification only, the following key-value pairs apply:
 - VERIFICATION_STATUS (the result of digits verification) is set to 1 if verification succeed, 0 if it did not.
 - ATTEMPTS is set to the number of digit-collection attempts made.
- For **TreatmentType**=RecordUserAnnouncement, the following key-value pair applies:
 - USER_ANN_ID is set to the message identifier, an integer, recorded by the user specified with USER_ID.

Event attribute	Short description
Event attribute	Short description

"> CallID (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> CallThreadID (mandatory) The call thread identifier of the call."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call to which this event relates."> CustomerID (mandatory) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> TreatmentType (mandatory) The type of treatment to be applied to the telephony object in question."> CollectedDigits (optional) A pointer to the digits that have been collected from the calling party."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> LastCollectedDigit (optional) The last digit collected from the calling party."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices."> TransferConnID (optional) Contains the **ConnID** of the call for which transfer or conference was initiated."> TreatmentParms (optional) Contains parameters to be used for the treatment."> UserData (optional) Specifies the pointer to the call-related user data.

EventTreatmentNotApplied

The call has not been treated for some reason. The reason is returned in **ErrorCode** and **ErrorMessage** parameters.

Event attribute	Short description
Event attribute	Short description

"> CallID (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> CallThreadID (mandatory) The call thread identifier of the call."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call to which this event relates."> CustomerID (mandatory) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> ErrorCode (mandatory) This attribute contains a value that indicates why a client request failed."> id (mandatory) The event name."> ReferenceID (mandatory) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> TreatmentType (mandatory) The type of treatment to be applied to the telephony object in question."> ErrorMessage (optional) A pointer to the character string containing additional information about an error."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> TransferConnID (optional) Contains the **ConnID** of the call for which transfer or conference was initiated."> TreatmentParms (optional) Contains parameters to be used for the treatment."> UserData (optional) Specifies the pointer to the call-related user data.

[>> Back to list](#)

Call routing events

Contents

- [1 EventRouteRequest](#)
- [2 EventRouteUsed](#)

Find information about the call routing events that Voice Microservices produce.

Related documentation:

-
-
-

RSS:

- [For private edition](#)

Call routing events are stored in the voice-callthread topic.

Voice Microservices support the following call routing events, which are described on this page. Click the attribute names in the tables below to see fuller descriptions of the attributes.

- [EventRouteRequest](#)
- [EventRouteUsed](#)

EventRouteRequest

The call has been placed on the routing point specified by **ThisDN**, and the switch is waiting for routing instructions.

The **PreviousConnID** attribute must appear if a call with **CallType**=Consult has been placed on a routing point.

ThisDN and **ThisQueue** attributes must have equal values.

Event attribute	Short description
Event attribute	Short description

"> CallID (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> CallThreadID (mandatory) The call thread identifier of the call."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call to which this event relates."> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group

or trunk group) with respect to the event in question."> ThisQueue (mandatory) The directory number of the most significant ACD group with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> ANI (optional) Automatic Number Identification. Indicates the telephony-company charge number."> CallHistory (optional) Information about transferring/routing of the call through a multisite contact center network."> CollectedDigits (optional) A pointer to the digits that have been collected from the calling party."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> DNIS (optional) The directory number to which the inbound call has been made."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> LastCollectedDigit (optional) The last digit collected from the calling party."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> OtherDNRole (optional) The role of the telephony object specified by **OtherDN** in the event in question."> OtherDN (optional) The directory number of the second most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> OtherQueue (optional) The directory number of the second most significant ACD group with respect to the event in question."> OtherTrunk (optional) The identifier of the second most significant trunk group with respect to the event in question."> PreviousConnID (optional) This attribute links two associated calls."> ThirdPartyDN (optional) The directory number of the third most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> ThisTrunk (optional) The identifier of the most significant trunk with respect to the event in question."> UserData (optional) Specifies the pointer to the call-related user data.

[>> Back to list](#)

EventRouteUsed

The call has been routed as requested in the function `TRouteCall()` or has been default routed by the switch after the routing timeout has expired (that is, there was no routing instruction from the computer domain within the specified timeout).

The **ThirdPartyDN** attribute specifies the destination DN or dialing number. It is:

- Mandatory if routing was done by Voice Microservices.
- Absent if the call was rejected. Optional in other cases.

For the EventRouteUsed event, **ThirdPartyDNRole**=Destination.

The **OtherDN** attribute is used to specify the target party when the forward feature is in progress.

ThisDN and **ThisQueue** attributes must have equal values.

Event attribute	Short description
Event attribute	Short description

"> CallID (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> CallThreadID (mandatory) The call thread identifier of the call."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call to which this event relates."> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> ThisQueue (mandatory) The directory number of the most significant ACD group with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> ANI (optional) Automatic Number Identification. Indicates the telephony-company charge number."> CallHistory (optional) Information about transferring/routing of the call through a multisite contact center network."> CallState (optional) The current status of the call the event relates to."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> DNIS (optional) The directory number to which the inbound call has been made."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> OtherDN (optional) The directory number of the second most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request

a client sends to Voice Microservices."> ThirdPartyDNRole (optional) The role of the telephony object specified by **ThirdPartyDN** in the event in question."> ThirdPartyDN (optional) The directory number of the third most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> ThisTrunk (optional) The identifier of the most significant trunk with respect to the event in question."> UserData (optional) Specifies the pointer to the call-related user data.

[>> Back to list](#)

Call handling and transfer/conference events

Contents

- [1 EventAbandoned](#)
- [2 EventDestinationBusy](#)
- [3 EventDialing](#)
- [4 EventDiverted](#)
- [5 EventEstablished](#)
- [6 EventHeld](#)
- [7 EventNetworkReached](#)
- [8 EventPartyAdded](#)
- [9 EventPartyChanged](#)
- [10 EventPartyDeleted](#)
- [11 EventQueued](#)
- [12 EventReleased](#)
- [13 EventRetrieved](#)
- [14 EventRinging](#)

Find information about the call handling and transfer/conference events that Voice Microservices produce.

Related documentation:

-
-
-

RSS:

- [For private edition](#)

Call handling and transfer/conference events are stored in the voice-callthread topic.

Voice Microservices support the following call handling and transfer/conference events, which are described on this page. Click the attribute names in the tables below to see fuller descriptions of the attributes.

- [EventAbandoned](#)
- [EventDestinationBusy](#)
- [EventDialing](#)
- [EventDiverted](#)
- [EventEstablished](#)
- [EventHeld](#)
- [EventNetworkReached](#)
- [EventPartyAdded](#)
- [EventPartyChanged](#)
- [EventPartyDeleted](#)
- [EventQueued](#)
- [EventReleased](#)
- [EventRetrieved](#)
- [EventRinging](#)

EventAbandoned

The caller abandoned the call before it was answered.

The **PreviousConnID** attribute must appear if the value of **CallType** is Consult.

The **ThisQueue** attribute must appear for an ACD call.

Event attribute	Short description
Event attribute	Short description

"> CallID (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> CallState (mandatory) The current status of the call the event relates to."> CallThreadID (mandatory) The call

thread identifier of the call."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call to which this event relates."> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDNRole (mandatory) The role of the telephony object specified by **ThisDN** in the event in question."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> ANI (optional) Automatic Number Identification. Indicates the telephony-company charge number."> CallHistory (optional) Information about transferring/routing of the call through a multisite contact center network."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> DNIS (optional) The directory number to which the inbound call has been made."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> OtherDNRole (optional) The role of the telephony object specified by **OtherDN** in the event in question."> OtherDN (optional) The directory number of the second most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> OtherQueue (optional) The directory number of the second most significant ACD group with respect to the event in question."> OtherTrunk (optional) The identifier of the second most significant trunk group with respect to the event in question."> PreviousConnID (optional) This attribute links two associated calls."> ThisQueue (optional) The directory number of the most significant ACD group with respect to the event in question."> ThisTrunk (optional) The identifier of the most significant trunk with respect to the event in question."> UserData (optional) Specifies the pointer to the call-related user data.

[>> Back to list](#)

EventDestinationBusy

The called party specified by **OtherDN** is busy with another call.

For scenarios initiated with `RequestMakeCall`, the **CallState** attribute might have values that clarify the reason for the destination being busy, for instance `CallStateSitInvalidNum`.

The **PreviousConnID** attribute must be included if the value of **CallType** is `Consult`.

Event attribute	Short description
Event attribute	Short description

"> **CallID** (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> **CallThreadID** (mandatory) The call thread identifier of the call."> **CallType** (mandatory) The type of call in question."> **ConnID** (mandatory) A current connection identifier of the call to which this event relates."> **id** (mandatory) The event name."> **Server** (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> **ThisDNRole** (mandatory) The role of the telephony object specified by **ThisDN** in the event in question."> **ThisDN** (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> **time** (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> **ANI** (optional) Automatic Number Identification. Indicates the telephony-company charge number."> **CallHistory** (optional) Information about transferring/routing of the call through a multisite contact center network."> **CallState** (optional) The current status of the call the event relates to."> **CustomerID** (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> **DNIS** (optional) The directory number to which the inbound call has been made."> **Extensions** (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> **NetworkCallID** (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> **NetworkNodeID** (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> **OtherDN** (optional) The directory number of the second most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> **OtherQueue** (optional) The directory number of the second most significant ACD group with respect to the event in question."> **OtherTrunk** (optional) The identifier of the second most significant trunk group with respect to the event in question."> **PreviousConnID** (optional) This attribute links two associated calls."> **ThisQueue** (optional) The directory number of the most significant ACD group with respect to the event in question."> **UserData** (optional) Specifies the pointer to the call-related user data.

EventDialing

An attempt to make a call on behalf of the telephony object specified by **ThisDN** is in progress.

CallType can be Unknown.

OtherDN is either a dialed number or not present if Voice Microservices have no information about the other party. **OtherDNRole** appears if the attribute **OtherDN** is present.

The **PreviousConnID** attribute must appear if the value of **CallType** is Consult.

ThisQueue must appear in predictive dialing and be equal to **ThisDN**.

Event attribute	Short description
Event attribute	Short description

"> CallID (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> CallThreadID (mandatory) The call thread identifier of the call."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call to which this event relates."> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDNRole (mandatory) The role of the telephony object specified by **ThisDN** in the event in question."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> CallHistory (optional) Information about transferring/routing of the call through a multisite contact center network."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> DNIS (optional) The directory number to which the inbound call has been made."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> OtherDNRole (optional) The role of the telephony object specified by **OtherDN** in the event in question."> OtherDN (optional) The directory number of the second most significant telephony object

(except an ACD group or trunk group) with respect to the event in question."> OtherQueue (optional) The directory number of the second most significant ACD group with respect to the event in question."> OtherTrunk (optional) The identifier of the second most significant trunk group with respect to the event in question."> PreviousConnID (optional) This attribute links two associated calls."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices."> ThisQueue (optional) The directory number of the most significant ACD group with respect to the event in question."> UserData (optional) Specifies the pointer to the call-related user data.

[>> Back to list](#)

EventDiverged

The call has been diverted from the queue to another telephony object.

The **PreviousConnID** attribute must be included if the value of **CallType** is Consult.

The **ThirdPartyDN** and **ThirdPartyQueue** attributes must be present if the value of **CallState** is Redirected. In all other call scenarios, **ThirdPartyDN** must be present only if such information is provided by a CTI link.

ThisDN and **ThisQueue** attributes must have equal values.

Event attribute	Short description
Event attribute	Short description

"> CallID (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> CallState (mandatory) The current status of the call the event relates to."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call to which this event relates."> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDNRole (mandatory) The role of the telephony object specified by **ThisDN** in the event in question."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in

question."> ThisQueue (mandatory) The directory number of the most significant ACD group with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> CallHistory (optional) Information about transferring/routing of the call through a multisite contact center network."> CollectedDigits (optional) A pointer to the digits that have been collected from the calling party."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> OtherDNRole (optional) The role of the telephony object specified by **OtherDN** in the event in question."> OtherDN (optional) The directory number of the second most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> OtherQueue (optional) The directory number of the second most significant ACD group with respect to the event in question."> OtherTrunk (optional) The identifier of the second most significant trunk group with respect to the event in question."> PreviousConnID (optional) This attribute links two associated calls."> ThirdPartyDNRole (optional) The role of the telephony object specified by **ThirdPartyDN** in the event in question."> ThirdPartyDN (optional) The directory number of the third most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> ThirdPartyQueue (optional) The directory number of the third most significant ACD group with respect to the event in question."> ThisTrunk (optional) The identifier of the most significant trunk with respect to the event in question."> UserData (optional) Specifies the pointer to the call-related user data.

[>> Back to list](#)

EventEstablished

For the application associated with the calling party: the telephony object specified by **OtherDN** has answered (either the calling party answered or the switch simulated an answer if option auto-answer is set on the switch) and the connection has been established. For the application associated with the called party: the call associated with **ConnID** has been established.

The **PreviousConnID** attribute must appear if the value of **CallType** is Consult.

Event attribute	Short description
Event attribute	Short description

"> CallID (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> CallThreadID (mandatory) The call thread identifier of the call."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call to which this event relates."> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDNRole (mandatory) The role of the telephony object specified by **ThisDN** in the event in question."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> ANI (optional) Automatic Number Identification. Indicates the telephony-company charge number."> CallHistory (optional) Information about transferring/routing of the call through a multisite contact center network."> CallState (optional) The current status of the call the event relates to."> CollectedDigits (optional) A pointer to the digits that have been collected from the calling party."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> DNIS (optional) The directory number to which the inbound call has been made."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> OtherDNRole (optional) The role of the telephony object specified by **OtherDN** in the event in question."> OtherDN (optional) The directory number of the second most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> OtherQueue (optional) The directory number of the second most significant ACD group with respect to the event in question."> OtherTrunk (optional) The identifier of the second most significant trunk group with respect to the event in question."> PreviousConnID (optional) This attribute links two associated calls."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices."> ThisQueue (optional) The directory number of the most

significant ACD group with respect to the event in question."> UserData (optional) Specifies the pointer to the call-related user data.

[>> Back to list](#)

EventHeld

The call has been placed on hold.

The **PreviousConnID** attribute must be included if the value of **CallType** is Consult.

Event attribute	Short description
Event attribute	Short description

"> CallID (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> CallThreadID (mandatory) The call thread identifier of the call."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call to which this event relates."> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDNRole (mandatory) The role of the telephony object specified by **ThisDN** in the event in question."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> ANI (optional) Automatic Number Identification. Indicates the telephony-company charge number."> CallHistory (optional) Information about transferring/routing of the call through a multisite contact center network."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> DNIS (optional) The directory number to which the inbound call has been made."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> OtherDNRole (optional) The role of the telephony object specified by **OtherDN** in the event in question."> OtherDN (optional) The directory number of the second most significant telephony object

(except an ACD group or trunk group) with respect to the event in question."> OtherQueue (optional) The directory number of the second most significant ACD group with respect to the event in question."> OtherTrunk (optional) The identifier of the second most significant trunk group with respect to the event in question."> PreviousConnID (optional) This attribute links two associated calls."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices."> ThisQueue (optional) The directory number of the most significant ACD group with respect to the event in question."> UserData (optional) Specifies the pointer to the call-related user data.

[>> Back to list](#)

EventNetworkReached

The call has reached the public network interface.

The **PreviousConnID** attribute must be included if the value of **CallType** is Consult.

Event attribute	Short description
Event attribute	Short description

"> CallID (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> CallThreadID (mandatory) The call thread identifier of the call."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call to which this event relates."> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDNRole (mandatory) The role of the telephony object specified by **ThisDN** in the event in question."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> ANI (optional) Automatic Number Identification. Indicates the telephony-company charge number."> CallHistory (optional) Information about transferring/routing of the call through a multisite contact center network.">

CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> DNIS (optional) The directory number to which the inbound call has been made."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> OtherDNRole (optional) The role of the telephony object specified by **OtherDN** in the event in question."> OtherDN (optional) The directory number of the second most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> OtherTrunk (optional) The identifier of the second most significant trunk group with respect to the event in question."> PreviousConnID (optional) This attribute links two associated calls."> ThisTrunk (optional) The identifier of the most significant trunk with respect to the event in question."> UserData (optional) Specifies the pointer to the call-related user data.

[>> Back to list](#)

EventPartyAdded

One or more parties has been added to the call as a result of a conference.

Voice MCS always distributes EventPartyAdded for every party. If only one party is added (as in the case of a simple conference call), the corresponding telephony object is specified in **OtherDN**. Similarly, if more than one party is added to a call and there are multiple EventPartyAdded messages, one for each party on the consultation call joining the main call, then the corresponding telephony object is specified in **OtherDN**.

The **ThirdPartyDN** and **ThirdPartyDNRole** attributes are not present if the switch does not distribute them to Voice Microservices.

Event attribute	Short description
Event attribute	Short description

"> CallThreadID (mandatory) The call thread identifier of the call."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call to which this event relates."> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices.">

ThirdPartyDNRole (mandatory) The role of the telephony object specified by **ThirdPartyDN** in the event in question."> ThirdPartyDN (mandatory) The directory number of the third most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> ThisDNRole (mandatory) The role of the telephony object specified by **ThisDN** in the event in question."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> ANI (optional) Automatic Number Identification. Indicates the telephony-company charge number."> CallHistory (optional) Information about transferring/routing of the call through a multisite contact center network."> CallState (optional) The current status of the call the event relates to."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> DNIS (optional) The directory number to which the inbound call has been made."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> OtherDNRole (optional) The role of the telephony object specified by **OtherDN** in the event in question."> OtherDN (optional) The directory number of the second most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> OtherQueue (optional) The directory number of the second most significant ACD group with respect to the event in question."> OtherTrunk (optional) The identifier of the second most significant trunk group with respect to the event in question."> ThisQueue (optional) The directory number of the most significant ACD group with respect to the event in question."> ThisTrunk (optional) The identifier of the most significant trunk with respect to the event in question."> UserData (optional) Specifies the pointer to the call-related user data.

[>> Back to list](#)

EventPartyChanged

The telephony object specified by **OtherDN** has replaced the telephony object specified by **OtherDN**

in the previously received event; or the **PreviousConnID** of the call has been given a new value, **ConnID**.

The value of **CallState** can be either Transferred or Conferenced.

The **OtherDN**, **OtherDNRole**, **OtherTrunk**, **ThirdPartyDNRole** attributes must not appear if the **CallState** is Conferenced.

The **ThirdPartyDN** attribute is not present if the switch does not distribute it to Voice Microservices.

Event attribute	Short description
Event attribute	Short description

"> CallID (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> CallState (mandatory) The current status of the call the event relates to."> CallThreadID (mandatory) The call thread identifier of the call."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call to which this event relates."> id (mandatory) The event name."> PreviousConnID (mandatory) This attribute links two associated calls."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThirdPartyDNRole (mandatory) The role of the telephony object specified by **ThirdPartyDN** in the event in question."> ThirdPartyDN (mandatory) The directory number of the third most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> ThisDNRole (mandatory) The role of the telephony object specified by **ThisDN** in the event in question."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> ANI (optional) Automatic Number Identification. Indicates the telephony-company charge number."> CallHistory (optional) Information about transferring/routing of the call through a multisite contact center network."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> DNIS (optional) The directory number to which the inbound call has been made."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> OtherDNRole (optional) The role of the telephony object specified by **OtherDN** in the event in question."> OtherDN (optional) The directory number of the second most significant telephony object (except an ACD group or trunk group) with respect to the event in question.">

OtherTrunk (optional) The identifier of the second most significant trunk group with respect to the event in question."> ThisQueue (optional) The directory number of the most significant ACD group with respect to the event in question."> ThisTrunk (optional) The identifier of the most significant trunk with respect to the event in question."> UserData (optional) Specifies the pointer to the call-related user data.

[>> Back to list](#)

EventPartyDeleted

The telephony object specified by **OtherDN** has been deleted from the conference call in question.

The **CallState** attribute indicates whether a call is still considered as a conference (that is, the number of parties in the call is more than two).

Event attribute	Short description
Event attribute	Short description

"> CallID (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> CallState (mandatory) The current status of the call the event relates to."> CallThreadID (mandatory) The call thread identifier of the call."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call to which this event relates."> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDNRole (mandatory) The role of the telephony object specified by **ThisDN** in the event in question."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> ANI (optional) Automatic Number Identification. Indicates the telephony-company charge number."> CallHistory (optional) Information about transferring/routing of the call through a multisite contact center network."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> DNIS (optional) The directory number to which the inbound call has been made."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other

parameters in an event or a request."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> OtherDNRole (optional) The role of the telephony object specified by **OtherDN** in the event in question."> OtherDN (optional) The directory number of the second most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> OtherQueue (optional) The directory number of the second most significant ACD group with respect to the event in question."> OtherTrunk (optional) The identifier of the second most significant trunk group with respect to the event in question."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices."> ThirdPartyDNRole (optional) The role of the telephony object specified by **ThirdPartyDN** in the event in question."> ThirdPartyDN (optional) The directory number of the third most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> ThisQueue (optional) The directory number of the most significant ACD group with respect to the event in question."> ThisTrunk (optional) The identifier of the most significant trunk with respect to the event in question."> UserData (optional) Specifies the pointer to the call-related user data.

>> [Back to list](#)

EventQueued

The call has been queued in the ACD group specified by **ThisQueue**.

The **PreviousConnID** attribute must be included if the value of **CallType** is Consult.

ThisDN and **ThisQueue** attributes must have equal values.

Event attribute	Short description
Event attribute	Short description

"> CallID (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call

to which this event relates."> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDNRole (mandatory) The role of the telephony object specified by **ThisDN** in the event in question."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> ThisQueue (mandatory) The directory number of the most significant ACD group with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> ANI (optional) Automatic Number Identification. Indicates the telephony-company charge number."> CallHistory (optional) Information about transferring/routing of the call through a multisite contact center network."> CallState (optional) The current status of the call the event relates to."> CollectedDigits (optional) A pointer to the digits that have been collected from the calling party."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> DNIS (optional) The directory number to which the inbound call has been made."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> LastCollectedDigit (optional) The last digit collected from the calling party."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> OtherDNRole (optional) The role of the telephony object specified by **OtherDN** in the event in question."> OtherDN (optional) The directory number of the second most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> OtherQueue (optional) The directory number of the second most significant ACD group with respect to the event in question."> OtherTrunk (optional) The identifier of the second most significant trunk group with respect to the event in question."> PreviousConnID (optional) This attribute links two associated calls."> ThirdPartyDN (optional) The directory number of the third most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> ThisTrunk (optional) The identifier of the most significant trunk with respect to the event in question."> UserData (optional) Specifies the pointer to the call-related user data.

[>> Back to list](#)

EventReleased

The telephony object specified by **ThisDN** has disconnected or has been dropped from the call.

The **OtherDN**, **OtherDNRole**, **OtherQueue**, and **OtherTrunk** attributes do not appear if the release is from a conference. In all other call scenarios, the attribute must be present only if such information is provided by a CTI link.

The **PreviousConnID** attribute must be included if the value of **CallType** is Consult.

The appearance of **ThirdPartyDN** depends on the following conditions:

- If information about the new destination is available from the switch at the moment when EventReleased is generated, then **ThirdPartyDN** is mandatory. Or, if Voice Microservices have initiated a single-step transfer, redirection, or previously set the forwarding target, this attribute is also mandatory.
- If a call has gone through a single-step transfer, been redirected, or forwarded by another application (not the Voice Microservices in question), the **ThirdPartyDN** attribute is absent.

Event attribute	Short description
Event attribute	Short description

"> CallID (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> CallState (mandatory) The current status of the call the event relates to."> CallThreadID (mandatory) The call thread identifier of the call."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call to which this event relates."> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDNRole (mandatory) The role of the telephony object specified by **ThisDN** in the event in question."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> ANI (optional) Automatic Number Identification. Indicates the telephony-company charge number."> CallHistory (optional) Information about transferring/routing of the call through a multisite contact center network."> Cause (optional) For network calls, the reason for transitions to certain states — Routing and NoParty."> CollectedDigits (optional) A pointer to the digits that have been collected from the calling party."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> DNIS (optional) The directory number to which the inbound call has been made."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request.">

NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> **NetworkNodeID** (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> **OtherDNRole** (optional) The role of the telephony object specified by **OtherDN** in the event in question."> **OtherDN** (optional) The directory number of the second most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> **OtherQueue** (optional) The directory number of the second most significant ACD group with respect to the event in question."> **OtherTrunk** (optional) The identifier of the second most significant trunk group with respect to the event in question."> **PreviousConnID** (optional) This attribute links two associated calls."> **Reasons** (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of **ThisDN**."> **ReferenceID** (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a **TSetReferenceID()** function call and attached to the request a client sends to Voice Microservices."> **ThirdPartyDN** (optional) The directory number of the third most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> **ThisQueue** (optional) The directory number of the most significant ACD group with respect to the event in question."> **ThisTrunk** (optional) The identifier of the most significant trunk with respect to the event in question."> **UserData** (optional) Specifies the pointer to the call-related user data.

>> [Back to list](#)

EventRetrieved

The call has been retrieved from hold.

In all call scenarios, the **OtherDN**, **OtherDNRole**, **OtherQueue**, and **OtherTrunk** attributes must be present only if the information is provided by a CTI link.

The value for the **ThisDNRole** and **ThisQueue** attributes is the same as that for the events preceding **EventRetrieved** (**EventEstablished** and **EventRinging**) for the same call. For non-ACD calls, **ThisQueue** is not reported.

Event attribute	Short description
Event attribute	Short description

"> **CallID** (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> **CallState** (mandatory) The current

status of the call the event relates to."> CallThreadID (mandatory) The call thread identifier of the call."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call to which this event relates."> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDNRole (mandatory) The role of the telephony object specified by **ThisDN** in the event in question."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> ANI (optional) Automatic Number Identification. Indicates the telephony-company charge number."> CallHistory (optional) Information about transferring/routing of the call through a multisite contact center network."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> DNIS (optional) The directory number to which the inbound call has been made."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> OtherDNRole (optional) The role of the telephony object specified by **OtherDN** in the event in question."> OtherDN (optional) The directory number of the second most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> OtherQueue (optional) The directory number of the second most significant ACD group with respect to the event in question."> OtherTrunk (optional) The identifier of the second most significant trunk group with respect to the event in question."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices."> ThisQueue (optional) The directory number of the most significant ACD group with respect to the event in question."> ThisTrunk (optional) The identifier of the most significant trunk with respect to the event in question."> UserData (optional) Specifies the pointer to the call-related user data.

>> Back to list

EventRinging

A call has been delivered to the telephony object specified by **ThisDN**.

The **PreviousConnID** attribute must appear if the value of **CallType** is Consult.

The **ThisQueue** attribute must appear for an ACD call.

Event attribute	Short description
Event attribute	Short description

"> CallID (mandatory) This attribute contains the call identification provided by the switch, which uniquely identifies a call."> CallState (mandatory) The current status of the call the event relates to."> CallThreadID (mandatory) The call thread identifier of the call."> CallType (mandatory) The type of call in question."> ConnID (mandatory) A current connection identifier of the call to which this event relates."> id (mandatory) The event name."> Server (mandatory) A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices."> ThisDNRole (mandatory) The role of the telephony object specified by **ThisDN** in the event in question."> ThisDN (mandatory) The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> time (mandatory) The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour)."> ANI (optional) Automatic Number Identification. Indicates the telephony-company charge number."> CallHistory (optional) Information about transferring/routing of the call through a multisite contact center network."> Cause (optional) For network calls, the reason for transitions to certain states — Routing and NoParty."> CollectedDigits (optional) A pointer to the digits that have been collected from the calling party."> CustomerID (optional) A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated."> DNIS (optional) The directory number to which the inbound call has been made."> Extensions (optional) A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request."> NetworkCallID (optional) In the case of network routing, the call identifier assigned by the switch where the call initially arrived."> NetworkNodeID (optional) In the case of network routing, the identifier of the switch where the call initially arrived."> OtherDNRole (optional) The role of the telephony object specified by **OtherDN** in the event in question."> OtherDN (optional) The directory number of the second most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> OtherQueue (optional) The directory number of the second most significant ACD group with respect to the event in question."> OtherTrunk (optional) The identifier of the

second most significant trunk group with respect to the event in question."> PreviousConnID (optional) This attribute links two associated calls."> Reasons (optional) A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN."> ReferenceID (optional) (Use is internal to Voice Microservices.) **ReferenceID** is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices."> ThirdPartyDN (optional) The directory number of the third most significant telephony object (except an ACD group or trunk group) with respect to the event in question."> ThisQueue (optional) The directory number of the most significant ACD group with respect to the event in question."> UserData (optional) Specifies the pointer to the call-related user data.

[>> Back to list](#)

Event attributes

Contents

- 1 Agent ID
- 2 AgentWorkMode
- 3 ANI
- 4 CallHistory
- 5 CallID
- 6 CallState
- 7 CallThreadID
- 8 CallType
- 9 Cause
- 10 CollectedDigits
- 11 ConnID
- 12 CustomerID
- 13 DNIS
- 14 ErrorCode
- 15 ErrorMessage
- 16 Extensions
- 17 id
- 18 InfoStatus
- 19 LastCollectedDigit
- 20 MonitorNextCallType
- 21 NetworkCallID
- 22 NetworkNodeID
- 23 OtherDN
- 24 OtherDNRole
- 25 OtherQueue
- 26 OtherTrunk
- 27 PreviousConnID
- 28 Reasons

- 29 ReferenceID
- 30 Server
- 31 ThirdPartyDN
- 32 ThirdPartyDNRole
- 33 ThirdPartyQueue
- 34 ThisDN
- 35 ThisDNRole
- 36 ThisQueue
- 37 ThisTrunk
- 38 time
- 39 TransferConnID
- 40 TreatmentParms
- 41 TreatmentType
- 42 UserData

Learn about the attributes that make up the events.

Related documentation:

-
-
-

RSS:

- [For private edition](#)

- Agent ID
- AgentWorkMode
- ANI
- CallHistory
- CallID
- CallState
- CallThreadID
- CallType
- Cause
- CollectedDigits
- ConnID
- CustomerID
- DNIS
- ErrorCode
- ErrorMessage
- Extensions
- id
- InfoStatus
- LastCollectedDigit
- MonitorNextCallType
- NetworkCallID
- NetworkNodeID
- OtherDNRole
- OtherDN
- OtherQueue
- OtherTrunk
- PreviousConnID
- Reasons
- ReferenceID
- Server
- ThirdPartyDNRole
- ThirdPartyDN
- ThirdPartyQueue
- ThisDNRole
- ThisDN
- ThisQueue
- ThisTrunk
- time
- TransferConnID
- TreatmentParms
- TreatmentType
- UserData

Agent ID

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: This attribute is not mandatory for any events.• Optional: EventAgentLogin, EventAgentLogout
Description	This parameter uniquely identifies the ACD agent.

[>> Back to list](#)

AgentWorkMode

Events that use the attribute	<ul style="list-style-type: none"> • Mandatory: EventAgentReady • Optional: EventAgentNotReady
Description	This attribute indicates the agent/supervisor-related current work mode.

[>> Back to list](#)

ANI

Events that use the attribute	<ul style="list-style-type: none"> • Mandatory: This attribute is not mandatory for any events. • Optional: EventAbandoned, EventDestinationBusy, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest, EventRouteUsed
Description	Automatic Number Identification. Indicates the telephony-company charge number.

[>> Back to list](#)

CallHistory

Events that use the attribute	<ul style="list-style-type: none"> • Mandatory: This attribute is not mandatory for any events. • Optional: EventOffHook, EventOnHook, EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest, EventRouteUsed
Description	Information about transferring/routing of the call through a multisite contact center network. Typically used to keep track of a call in multisite contact centers.

[>> Back to list](#)

CallID

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest, EventRouteUsed, EventTreatmentApplied, EventTreatmentEnd, EventTreatmentNotApplied• Optional: EventOffHook, EventOnHook
Description	This attribute contains the call identification provided by the switch, which uniquely identifies a call. As opposed to ConnID that is assigned by Voice Microservices, CallID is created by the switch when the incoming call arrives, or when agent/system out-dial calls are created. The attribute must be present if the switch generates and distributes the corresponding parameter to Voice Microservices. (CallID is zero as long as the switch does not provide that information to Voice Microservices.)

[>> Back to list](#)

CallState

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: EventAbandoned, EventDiverted, EventPartyChanged, EventPartyDeleted, EventReleased, EventRetrieved, EventRinging• Optional: EventOffHook, EventDestinationBusy, EventEstablished, EventPartyAdded, EventQueued, EventRouteUsed
Description	The current status of the call the event relates to.

[>> Back to list](#)

CallThreadID

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: EventAbandoned, EventDestinationBusy, EventDialing, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventReleased, EventRetrieved, EventRinging, EventRouteRequest, EventRouteUsed, EventTreatmentApplied, EventTreatmentEnd, EventTreatmentNotApplied• Optional: This attribute is not optional for any events.
Description	The call thread identifier of the call.

[>> Back to list](#)

CallType

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest, EventRouteUsed, EventTreatmentApplied, EventTreatmentEnd, EventTreatmentNotApplied• Optional: EventOffHook
Description	The type of call in question.

[>> Back to list](#)

Cause

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: This attribute is not mandatory for any events.• Optional: EventReleased, EventRinging
Description	For network calls, the reason for transitions to certain states — Routing and NoParty. This helps clarify delivery failure, such as Busy or NoAnswer.

[>> Back to list](#)

CollectedDigits

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: This attribute is not mandatory for any events.• Optional: EventDiverted, EventEstablished, EventQueued, EventReleased, EventRinging, EventRouteRequest, EventTreatmentEnd
Description	A pointer to the digits that have been collected from the calling party.

[>> Back to list](#)

ConnID

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: EventMuteOff, EventMuteOn, EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest, EventRouteUsed, EventTreatmentApplied, EventTreatmentEnd, EventTreatmentNotApplied• Optional: EventOffHook, EventOnHook																																																																																										
Description	<p>A current connection identifier of the call to which this event relates.</p> <p>Connection ID structure</p> <table><tr><th>Byte</th><th colspan="8">Bits</th></tr><tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td>0</td><td colspan="2">Reserved</td><td colspan="6">Global Server Identifier</td></tr><tr><td>1</td><td colspan="8">Global Server Identifier</td></tr><tr><td>2</td><td colspan="8">Local Connection Identifier</td></tr><tr><td>3</td><td colspan="8">Local Connection Identifier</td></tr><tr><td>4</td><td colspan="8">Local Connection Identifier</td></tr><tr><td>5</td><td colspan="8">Local Connection Identifier</td></tr><tr><td>6</td><td colspan="8">Local Connection Identifier</td></tr><tr><td>7</td><td colspan="8">Local Connection Identifier</td></tr></table> <p>ConnID Parameters</p> <p>Reserved (bits 0 and 1): Bits reserved for future usage.</p> <p>Global Server Identifier (bits 2-15): 0 is a global server identifier.</p> <p>Local Connection Identifier (bits 16-63): Local identifier of the call this event relates to.</p>	Byte	Bits									0	1	2	3	4	5	6	7	0	Reserved		Global Server Identifier						1	Global Server Identifier								2	Local Connection Identifier								3	Local Connection Identifier								4	Local Connection Identifier								5	Local Connection Identifier								6	Local Connection Identifier								7	Local Connection Identifier							
Byte	Bits																																																																																										
	0	1	2	3	4	5	6	7																																																																																			
0	Reserved		Global Server Identifier																																																																																								
1	Global Server Identifier																																																																																										
2	Local Connection Identifier																																																																																										
3	Local Connection Identifier																																																																																										
4	Local Connection Identifier																																																																																										
5	Local Connection Identifier																																																																																										
6	Local Connection Identifier																																																																																										
7	Local Connection Identifier																																																																																										

>> Back to list

CustomerID

Events that use the attribute	<ul style="list-style-type: none"> Mandatory: EventTreatmentApplied, EventTreatmentEnd, EventTreatmentNotApplied Optional: EventAgentLogin, EventAgentLogout, EventAgentNotReady, EventAgentReady, EventDNDOff, EventDNDOn, EventForwardCancel, EventForwardSet, EventMuteOff, EventMuteOn, EventOffHook, EventOnHook, EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached,
--------------------------------------	---

	EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest, EventRouteUsed
Description	A pointer to the string containing the assigned Customer (Tenant) identifier through which the processing of the call was initiated. The attribute must be present in every event for a multitenant contact center.

[>> Back to list](#)

DNIS

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: This attribute is not mandatory for any events.• Optional: EventAbandoned, EventDestinationBusy, EventDialing, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest, EventRouteUsed
Description	Directory Number Information Service. The directory number to which the inbound call has been made.

[>> Back to list](#)

ErrorCode

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: EventTreatmentNotApplied• Optional: This attribute is not optional for any events.
Description	This attribute contains a value that indicates why a client request failed.

[>> Back to list](#)

ErrorMessage

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: This attribute is not mandatory for any events.• Optional: EventTreatmentNotApplied
--------------------------------------	---

Description	A pointer to the character string containing additional information about an error.
--------------------	---

[>> Back to list](#)

Extensions

Events that use the attribute	<ul style="list-style-type: none"> • Mandatory: EventTreatmentApplied • Optional: EventAgentLogin, EventAgentLogout, EventAgentNotReady, EventAgentReady, EventDNBackInService, EventDNDOff, EventDNDOn, EventDNOutOfService, EventForwardCancel, EventForwardSet, EventMuteOff, EventMuteOn, EventOffHook, EventOnHook, EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest, EventRouteUsed, EventTreatmentEnd, EventTreatmentNotApplied
Description	<p>A pointer to an additional data structure that takes into account switch-specific features that cannot be described by the other parameters in an event or a request. Extensions that are specific to particular events are noted with their event information in the Events section. Some extensions for requests permit tuning of Voice Microservices operations.</p> <p>If present, the Extensions attribute can include a ReasonCode value specifically used to communicate hardware reasons.</p>

[>> Back to list](#)

id

Events that use the attribute	<ul style="list-style-type: none"> • Mandatory: EventAgentLogin, EventAgentLogout, EventAgentNotReady, EventAgentReady, EventDNBackInService, EventDNDOff, EventDNDOn, EventDNOutOfService, EventForwardCancel, EventForwardSet, EventMuteOff, EventMuteOn, EventOffHook, EventOnHook, EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest, EventRouteUsed, EventTreatmentApplied, EventTreatmentEnd, EventTreatmentNotApplied • Optional: This attribute is not optional for any events.
Description	The event name.

[>> Back to list](#)

InfoStatus

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: This attribute is not mandatory for any events.• Optional: EventForwardSet
Description	The InfoType information about the telephony object specified by ThisDN and/or ThisQueue .

[>> Back to list](#)

LastCollectedDigit

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: This attribute is not mandatory for any events.• Optional: EventQueued, EventRouteRequest, EventTreatmentEnd
Description	The last digit collected from the calling party.

[>> Back to list](#)

MonitorNextCallType

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: This attribute is not mandatory for any events.• Optional: This attribute is not optional for any events.
Description	Indicates whether to monitor one (MonitorOneCall (0)) or all (MonitorAllCalls (1)) of the next calls.

[>> Back to list](#)

NetworkCallID

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: This attribute is not mandatory for any events.• Optional: EventMuteOff, EventMuteOn, EventOffHook, EventOnHook,
--------------------------------------	--

	EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest, EventRouteUsed, EventTreatmentApplied, EventTreatmentEnd, EventTreatmentNotApplied
Description	In the case of network routing, the call identifier assigned by the switch where the call initially arrived.

[>> Back to list](#)

NetworkNodeID

Events that use the attribute	<ul style="list-style-type: none"> • Mandatory: This attribute is not mandatory for any events. • Optional: EventMuteOff, EventMuteOn, EventOffHook, EventOnHook, EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest, EventRouteUsed, EventTreatmentApplied, EventTreatmentEnd, EventTreatmentNotApplied
Description	In the case of network routing, the identifier of the switch where the call initially arrived.

[>> Back to list](#)

OtherDN

Events that use the attribute	<ul style="list-style-type: none"> • Mandatory: This attribute is not mandatory for any events. • Optional: EventForwardSet, EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest, EventRouteUsed
Description	The directory number of the second most significant telephony object (except an ACD group or trunk group) with respect to the event in question. The application does not have to be registered to this directory number to receive the event in question.

[>> Back to list](#)

OtherDNRole

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: This attribute is not mandatory for any events.• Optional: EventAbandoned, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest
Description	The role of the telephony object specified by OtherDN in the event in question.

[>> Back to list](#)

OtherQueue

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: This attribute is not mandatory for any events.• Optional: EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventPartyAdded, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest
Description	The directory number of the second most significant ACD group with respect to the event in question.

[>> Back to list](#)

OtherTrunk

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: This attribute is not mandatory for any events.• Optional: EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest
Description	The identifier of the second most significant trunk group with respect to the event in question.

[>> Back to list](#)

PreviousConnID

Events that use the attribute	<ul style="list-style-type: none"> • Mandatory: EventPartyChanged • Optional: EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached, EventQueued, EventReleased, EventRinging, EventRouteRequest
Description	<p>This attribute links two associated calls. For example, events related to an original call include the connection ID of a consultation call; events related to a consultation call include the connection ID of the original call. For more information, see ConnID.</p> <p>WARNING: When EventPartyChanged is generated for the party that is still only involved in an original call (that is, ConnID has not been changed during a two-step operation), the PreviousConnID attribute is equal to ConnID of the original call.</p>

>> Back to list

Reasons

Events that use the attribute	<ul style="list-style-type: none"> • Mandatory: This attribute is not mandatory for any events. • Optional: EventAgentLogin, EventAgentLogout, EventAgentNotReady, EventAgentReady, EventDNDOff, EventDNDOOn, EventForwardCancel, EventForwardSet, EventMuteOff, EventMuteOn, EventDialing, EventEstablished, EventHeld, EventPartyDeleted, EventReleased, EventRetrieved, EventRinging, EventRouteUsed, EventTreatmentApplied, EventTreatmentEnd, EventTreatmentNotApplied
Description	<p>A pointer to an additional data structure that provides reasons for and results of actions taken by the user of ThisDN. Any Reasons attribute that appears in an event is taken directly from the corresponding request (see ReferenceID in events that correspond to requests). There is no other source for the information found in the content of the Reasons attribute. That is, no Reasons attribute should be expected for an event that is unsolicited. An event with no reference ID has no identifiable request that prompted it.</p> <p>Switch information of a similar nature to the Genesys Reasons attribute is sometimes available, but those switch reasons are passed in the Extensions attribute.</p>

>> Back to list

ReferenceID

Events that use the attribute	<ul style="list-style-type: none"> • Mandatory: EventTreatmentNotApplied • Optional: EventAgentLogin, EventAgentLogout, EventAgentNotReady, EventAgentReady, EventDNDOff, EventDNDOOn, EventForwardCancel, EventForwardSet, EventMuteOff, EventMuteOn, EventDialing, EventEstablished, EventHeld, EventPartyDeleted, EventReleased, EventRetrieved, EventRinging, EventRouteUsed, EventTreatmentApplied, EventTreatmentEnd 																																						
Description	<p>(Use is internal to Voice Microservices.) ReferenceID is the identifier generated by Voice Microservices or a TSetReferenceID() function call and attached to the request a client sends to Voice Microservices. Every time a client sends a request to Voice Microservices, it uses the current ReferenceID (increasing it by one each time). In response, Voice Microservices generate an event. The resulting event includes the same ReferenceID that was attached to the request only in the response to the client who initiated the request, as acknowledgment that the request has been fulfilled. If the request fails, EventError is sent only to the requestor.</p> <p>For more information, see the following table, which lists the events in which you will find the ReferenceID corresponding to that found with the request that prompted its assignment initially.</p> <p>ReferenceID in events that correspond to requests</p> <table border="1"> <thead> <tr> <th>Request</th><th>Event</th></tr> </thead> <tbody> <tr> <td colspan="2">General Requests</td></tr> <tr> <td>TOpenServer</td><td>Not Applicable</td></tr> <tr> <td>TOpenServerEx</td><td>Not Applicable</td></tr> <tr> <td>TDispatch</td><td>Not Applicable</td></tr> <tr> <td>TCloseServer</td><td>Not Applicable</td></tr> <tr> <td>TScanServer</td><td>Not Applicable</td></tr> <tr> <td>TScanServerEx</td><td>Not Applicable</td></tr> <tr> <td>TSetInputMask</td><td>EventACK</td></tr> <tr> <td colspan="2">Registration Requests</td></tr> <tr> <td>TRegisterAddress ^a</td><td>EventRegistered</td></tr> <tr> <td>TUnregisterAddress ^a</td><td>EventUnregistered</td></tr> <tr> <td colspan="2">Call Handling Requests</td></tr> <tr> <td>TAnswerCall</td><td>EventEstablished</td></tr> <tr> <td>TClearCall</td><td>EventReleased</td></tr> <tr> <td>THoldCall</td><td>EventHeld</td></tr> <tr> <td>TMakeCall</td><td>EventDialing</td></tr> <tr> <td>TMakePredictiveCall</td><td>EventDialing</td></tr> <tr> <td>TReleaseCall</td><td>EventReleased</td></tr> </tbody> </table>	Request	Event	General Requests		TOpenServer	Not Applicable	TOpenServerEx	Not Applicable	TDispatch	Not Applicable	TCloseServer	Not Applicable	TScanServer	Not Applicable	TScanServerEx	Not Applicable	TSetInputMask	EventACK	Registration Requests		TRegisterAddress ^a	EventRegistered	TUnregisterAddress ^a	EventUnregistered	Call Handling Requests		TAnswerCall	EventEstablished	TClearCall	EventReleased	THoldCall	EventHeld	TMakeCall	EventDialing	TMakePredictiveCall	EventDialing	TReleaseCall	EventReleased
Request	Event																																						
General Requests																																							
TOpenServer	Not Applicable																																						
TOpenServerEx	Not Applicable																																						
TDispatch	Not Applicable																																						
TCloseServer	Not Applicable																																						
TScanServer	Not Applicable																																						
TScanServerEx	Not Applicable																																						
TSetInputMask	EventACK																																						
Registration Requests																																							
TRegisterAddress ^a	EventRegistered																																						
TUnregisterAddress ^a	EventUnregistered																																						
Call Handling Requests																																							
TAnswerCall	EventEstablished																																						
TClearCall	EventReleased																																						
THoldCall	EventHeld																																						
TMakeCall	EventDialing																																						
TMakePredictiveCall	EventDialing																																						
TReleaseCall	EventReleased																																						

TRetrieveCall	EventRetrieved
TRedirectCall	EventReleased
Transfer/Conference Requests	
TInitiateConference	EventDialing
TInitiateTransfer	EventDialing
TCompleteConference	EventReleased
TCompleteTransfer	First arriving EventReleased
TDeleteFromConference	EventPartyDeleted or EventReleased
TReconnectCall	EventRetrieved
TMergeCalls	EventReleased
TMuteTransfer	EventDialing
TAlternateCall	EventHeld
TSingleStepConference	EventPartyAdded or EventRinging
TSingleStepTransfer	EventReleased
Call-Routing Requests	
TRouteCall	EventRouteUsed
Call Treatment Requests	
TApplyTreatment	EventTreatmentApplied+ EventTreatmentEnd or EventTreatmentNotApplied
TGiveMusicTreatment	EventTreatmentApplied
TGiveRingBackTreatment	EventTreatmentApplied
TGiveSilenceTreatment	EventTreatmentApplied
DTMF Requests	
TCollectDigits	EventDigitsCollected
TSendDTMF	EventDTMFSent
Voice-Mail Requests	
TOpenVoiceFile	EventVoiceFileOpened
TCloseVoiceFile	EventVoiceFileClosed
TLoginMailBox	EventMailBoxLogin
TLogoutMailBox	EventMailBoxLogout
TPlayVoice	EventVoiceFileEndPlay
Agent and DN Feature Requests	
TAgentLogin	EventAgentLogin
TAgentLogout	EventAgentLogout or EventQueueLogout
TAgentSetReady	EventAgentReady

TAgentSetNotReady	EventAgentNotReady
TCallSetForward	EventForwardSet
TCallCancelForward	EventForwardCancel
TMonitorNextCall	EventMonitoringNextCall
TCancelMonitoring	EventMonitoringCancelled
TSetMuteOff	EventMuteOff
TSetMuteOn	EventMuteOn
TListenDisconnect	EventListenDisconnected
TListenReconnect	EventListenReconnected
TSetDNDOOn	EventDNDOOn
TSetDNDOff	EventDNDOff
TSetMessageWaitingOn	EventMessageWaitingOn
TSetMessageWaitingOff	EventMessageWaitingOff
Query Requests	
TQueryAddress ^a	EventAddressInfo
TQueryCall ^a	EventPartyInfo
TQueryLocation ^a	EventLocationInfo
TQueryServer ^a	EventServerInfo
TQuerySwitch ^a	EventSwitchInfo
User-Data Requests	
TAttachUserData	EventAttachedDataChanged
TUpdateUserData	EventAttachedDataChanged
TDeleteUserData	EventAttachedDataChanged
TDeleteAllUserData	EventAttachedDataChanged
Special Requests	
TReserveAgent	EventAgentReserved
TSendUserEvent	EventACK
TSendEvent	EventACK
TSendEventEx	EventACK
TSetCallAttributes	EventCallInfoChanged
TPrivateService	EventPrivateInfo or EventAck

^a Only the requestor will receive a notification of the event associated with this request.

[>> Back to list](#)

Server

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: EventAgentLogin, EventAgentLogout, EventAgentNotReady, EventAgentReady, EventDNDOff, EventDNDOn, EventForwardCancel, EventForwardSet, EventOffHook, EventOnHook, EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest, EventRouteUsed, EventTreatmentApplied, EventTreatmentEnd, EventTreatmentNotApplied• Optional: This attribute is not optional for any events.
Description	A unique identifier assigned by Voice Microservices to the connection between a client and Voice Microservices.

[>> Back to list](#)

ThirdPartyDN

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: EventPartyAdded, EventPartyChanged• Optional: EventDiverted, EventPartyDeleted, EventQueued, EventReleased, EventRinging, EventRouteRequest, EventRouteUsed
Description	The directory number of the third most significant telephony object (except an ACD group or trunk group) with respect to the event in question. The application does not have to be registered to this directory number to receive the event in question.

[>> Back to list](#)

ThirdPartyDNRole

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: EventPartyAdded, EventPartyChanged• Optional: EventDiverted, EventPartyDeleted, EventRouteUsed
Description	The role of the telephony object specified by ThirdPartyDN in the event in question.

[>> Back to list](#)

ThirdPartyQueue

Events that use the attribute	<ul style="list-style-type: none"> • Mandatory: This attribute is not mandatory for any events. • Optional: EventDiverted
Description	The directory number of the third most significant ACD group with respect to the event in question.

[>> Back to list](#)

ThisDN

Events that use the attribute	<ul style="list-style-type: none"> • Mandatory: EventAgentLogin, EventAgentLogout, EventAgentNotReady, EventAgentReady, EventDNBackInService, EventDNDOff, EventDNDOn, EventDNOutOfService, EventForwardCancel, EventForwardSet, EventMuteOff, EventMuteOn, EventOffHook, EventOnHook, EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest, EventRouteUsed, EventTreatmentApplied, EventTreatmentEnd, EventTreatmentNotApplied • Optional: This attribute is not optional for any events.
Description	The directory number of the most significant telephony object (except an ACD group or trunk group) with respect to the event in question. The application must be registered to this directory number to receive the event in question.

[>> Back to list](#)

ThisDNRole

Events that use the attribute	<ul style="list-style-type: none"> • Mandatory: EventMuteOff, EventMuteOn, EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventTreatmentApplied • Optional: EventOnHook
Description	The role of the telephony object specified by ThisDN in the event in question.

[>> Back to list](#)

ThisQueue

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: EventDiverted, EventQueued, EventRouteRequest, EventRouteUsed• Optional: EventAgentLogin, EventAgentLogout, EventAgentNotReady, EventAgentReady, EventAbandoned, EventDestinationBusy, EventDialing, EventEstablished, EventHeld, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventReleased, EventRetrieved, EventRinging
Description	The directory number of the most significant ACD group with respect to the event in question.

[>> Back to list](#)

ThisTrunk

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: This attribute is not mandatory for any events.• Optional: EventOffHook, EventOnHook, EventAbandoned, EventDiverted, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRouteRequest, EventRouteUsed
Description	The identifier of the most significant trunk with respect to the event in question.

[>> Back to list](#)

time

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: EventAgentLogin, EventAgentLogout, EventAgentNotReady, EventAgentReady, EventDNDOff, EventDNDOOn, EventForwardCancel, EventForwardSet, EventOffHook, EventOnHook, EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest, EventRouteUsed, EventTreatmentApplied, EventTreatmentEnd, EventTreatmentNotApplied
--------------------------------------	--

	<ul style="list-style-type: none">• Optional: This attribute is not optional for any events.
Description	The structure specifies event generation time that is expressed in elapsed seconds and microseconds since 00:00 GMT, January 1, 1970 (zero hour).

[>> Back to list](#)

TransferConnID

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: This attribute is not mandatory for any events.• Optional: EventTreatmentApplied, EventTreatmentEnd, EventTreatmentNotApplied
Description	Contains the ConnID of the call for which transfer or conference was initiated.

[>> Back to list](#)

TreatmentParms

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: This attribute is not mandatory for any events.• Optional: EventTreatmentApplied, EventTreatmentEnd, EventTreatmentNotApplied
Description	Contains parameters to be used for the treatment.

[>> Back to list](#)

TreatmentType

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: EventTreatmentApplied, EventTreatmentEnd, EventTreatmentNotApplied• Optional: This attribute is not optional for any events.
Description	The type of treatment to be applied to the telephony object in question.

[>> Back to list](#)

UserData

Events that use the attribute	<ul style="list-style-type: none">• Mandatory: This attribute is not mandatory for any events.• Optional: EventMuteOff, EventMuteOn, EventOffHook, EventOnHook, EventAbandoned, EventDestinationBusy, EventDialing, EventDiverted, EventEstablished, EventHeld, EventNetworkReached, EventPartyAdded, EventPartyChanged, EventPartyDeleted, EventQueued, EventReleased, EventRetrieved, EventRinging, EventRouteRequest, EventRouteUsed, EventTreatmentApplied, EventTreatmentEnd, EventTreatmentNotApplied
Description	Specifies the pointer to the call-related user data.

[>> Back to list](#)

Basic call models

Contents

- 1 Simple call model
- 2 Connection-establishing phase for an internal/inbound call
 - 2.1 Abnormal call flow
- 3 Connection-establishing phase for an internal/inbound call to ACD
 - 3.1 Abnormal call flow
- 4 Connection-establishing phase for an internal/inbound call queued to multiple ACDs
 - 4.1 Abnormal call flow
- 5 Connection-establishing phase for an internal/inbound call with call parking
 - 5.1 Abnormal call flow
- 6 Connection-establishing phase for internal/inbound call with routing (RouteQueue case)
 - 6.1 Abnormal call flow
- 7 Connection-establishing phase for internal/inbound call with routing
 - 7.1 Abnormal call flow
- 8 Connection-establishing phase for an internal/inbound call with routing outbound
 - 8.1 Abnormal call flow
- 9 Connection-establishing phase for an outbound call
 - 9.1 Abnormal call flow
- 10 Connection-establishing phase while on hold (internal/outbound call)

This page describes the basic scenarios in which calls arrive in a contact center.

Related documentation:

-
-
-

RSS:

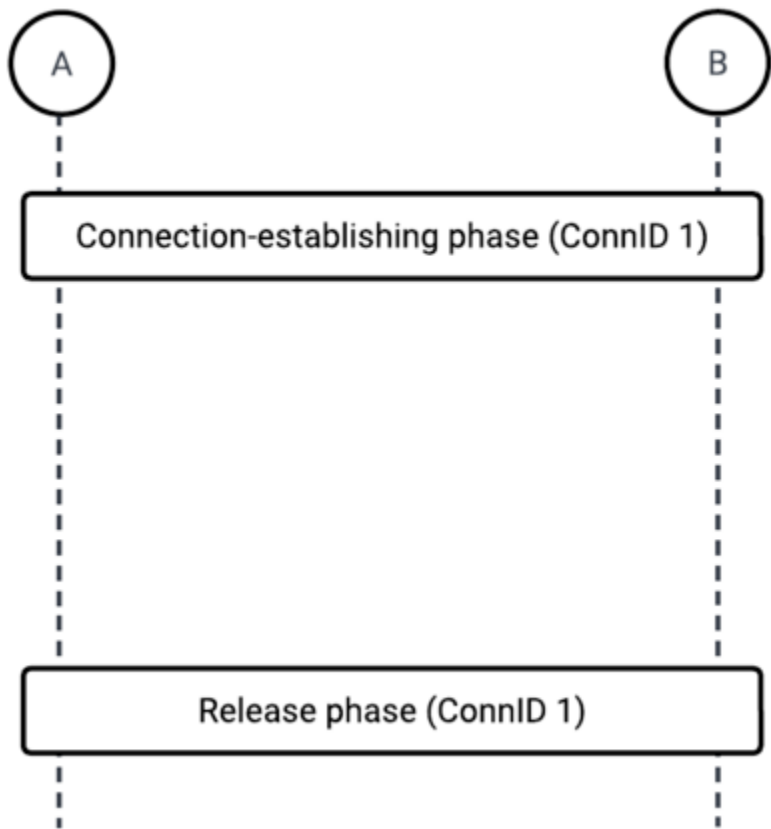
- [For private edition](#)

For simplicity, the examples on this page use abbreviated attribute values. For example, ConnID **1**, which – in actual events – displays as ConnID>@metainformation>.

The following comments and abbreviations are used in the call models:

- OPT—Optional.
- DIAL—Might be a dialed number or is not present if Voice Microservices have no information about the other party.

Simple call model



Connection-establishing phase for an internal/inbound call

The following table describes the connection-establishing phase for an internal/inbound call.

Party A	Party B
Make call to B (TMakeCall)	
EventDialing ConnID 1 ThisDN A ThisDNRole Origination OtherDN *DIAL OtherDNRole Destination *DIAL	
	EventRinging ConnID 1 ThisDN B ThisDNRole Destination

	OtherDN A OtherDNRole Origination CallState OK
	Answer (TAnswerCall)
EventEstablished ConnID 1 ThisDN A ThisDNRole Origination OtherDN B OtherDNRole Destination	EventEstablished ConnID 1 ThisDN B ThisDNRole Destination OtherDN A OtherDNRole Origination
Conversation	

Abnormal call flow

Interruption point	Party A	Party B
*	EventReleased ConnID 1 ThisDN A ThisDNRole Origination CallState OK	
**	EventDestinationBusy ConnID 1 ThisDN A ThisDNRole Origination CallState ^a	
***	EventReleased ConnID 1 ThisDN A ThisDNRole Origination OtherDN B ^{*DIAL} OtherDNRole Destination ^{*DIAL} CallState OK	EventAbandoned ConnID 1 ThisDN B OtherDN A CallState OK

a. CallState might have values that clarify the reason for the destination being busy, for instance CallState SitInvalidNum.

Connection-establishing phase for an internal/inbound call to ACD

The following table describes the connection-establishing phase for an internal/inbound call to ACD.

Party A	Party B (ACD Group)	Party C
Make call to B		

EventDialing ConnID 1 ThisDN A ThisDNRole Origination OtherDN B ^{*DIAL} OtherDNRole Destination ^{*DIAL}	EventQueued ConnID 1 ThisDN B ThisQueue B ThisDNRole Destination OtherDN A OtherDNRole Origination	
	Diverts call to C	
	EventDiverted ConnID 1 ThisDN B ThisQueue B ThisDNRole Destination OtherDN A OtherDNRole Origination ThirdPartyDN C ^{*OPT} ThirdPartyDNRole Destination ^{*OPT}	
		EventRinging ConnID 1 ThisDN C ThisQueue B ThisDNRole Destination OtherDN A OtherDNRole Origination CallState OK
		Answer (TAnswerCall)
EventEstablished ConnID 1 ThisDN A ThisDNRole Origination OtherDN C OtherDNRole Destination		EventEstablished ConnID 1 ThisDN C ThisQueue B ThisDNRole Destination OtherDN A OtherDNRole Origination
Conversation		

Abnormal call flow

Interruption point	Party A	Party B	Party C
*	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventAbandoned ConnID 1 ThisDN B OtherDN A CallState OK	
**	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK		
***	EventReleased		EventAbandoned

	ConnID 1 ThisDN A OtherDN C CallState OK		ConnID 1 ThisDN C OtherDN A CallState OK
--	---	--	---

Connection-establishing phase for an internal/inbound call queued to multiple ACDs

The following table describes the connection-establishing phase for an internal/inbound call queued to multiple ACDs.

Party A	Party B (ACD)	Party C (ACD)	Party D
Make internal/inbound call to B (ACD)			
EventDialing ConnID 1 ThisDN A ThisDNRole Origination OtherDN B *DIAL OtherDNRole Destination *DIAL	EventQueued ConnID 1 ThisDN B ThisQueue B ThisDNRole Destination OtherDN A OtherDNRole Origination		
		EventQueued ConnID 1 ThisDN C ThisQueue C ThisDNRole Destination OtherDN A OtherDNRole Origination	
	Diverts call to D		
	EventDiverted ConnID 1 ThisDN B ThisDNRole Origination OtherDN C OtherDNRole Destination	EventDiverted ConnID 1 ThisDN C ThisQueue C ThirdPartyDN D ThirdPartyQueue B CallState Redirected ^a	
			EventRinging ConnID 1 ThisDN D ThisQueue B ThisDNRole Destination OtherDN A OtherDNRole Origination CallState OK
			Answer (TAnswerCall)

EventEstablished ConnID 1 ThisDN A ThisDNRole Origination OtherDN D OtherDNRole Destination CallState OK			EventEstablished ConnID 1 ThisDN D ThisDNRole Destination OtherDN A OtherDNRole Origination CallState OK
Conversation			

a. For ACD configurations where calls are distributed to agents assigned directly to ACD groups, CallState with a value of Redirected is present. For ACD configurations where calls are distributed to agents assigned to secondary ACD groups associated with top-level ACD queues, the CallState, with the value Redirected, is not present.

Abnormal call flow

Interruption point	Party A	Party B	Party C	Party D
*	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventAbandoned ConnID 1 ThisDN B ThisQueue B OtherDN A CallState OK		
**	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventAbandoned ConnID 1 ThisDN B ThisQueue B OtherDN A CallState OK	EventAbandoned ConnID 1 ThisDN C ThisQueue C OtherDN A CallState OK	
***	EventReleased ConnID 1 ThisDN A OtherDN D CallState OK			
****	EventReleased ConnID 1 ThisDN A OtherDN D CallState OK			EventAbandoned ConnID 1 ThisDN D ThisQueue C OtherDN A CallState OK

Connection-establishing phase for an internal/inbound call with call parking

The following table describes the connection-establishing phase for an internal/inbound call with call

parking.

Party A	Party B
Make call to B (TMakeCall)	
EventDialing ConnID 1 ThisDN A ThisDNRole Origination OtherDN B *DIAL OtherDNRole Destination *DIAL	
	Call is parked on B
EventDestinationBusy *OPT ConnID 1 ThisDN A ThisDNRole Origination OtherDN B *DIAL OtherDNRole Destination *DIAL	EventQueued ConnID 1 ThisDN B ThisDNRole Destination OtherDN A OtherDNRole Origination CallState OK
	Call is picked up by B
	EventRinging ConnID 1 ThisDN B ThisDNRole Destination OtherDN A OtherDNRole Origination CallState OK
	Answer (TAnswerCall)
EventEstablished ConnID 1 ThisDN A ThisDNRole Origination OtherDN B OtherDNRole Destination	EventEstablished ConnID 1 ThisDN B ThisDNRole Destination OtherDN A OtherDNRole Origination
Conversation	

Abnormal call flow

Interruption point	Party A	Party B
*	EventReleased ConnID 1 ThisDN A ThisDNRole Origination OtherDN B *DIAL OtherDNRole Destination *DIAL CallState OK	EventAbandoned ConnID 1 ThisDN B OtherDN A CallState OK

Connection-establishing phase for internal/inbound call with routing (**RouteQueue** case)

The following table describes the connection-establishing phase for an internal/inbound call with routing (**RouteQueue** case).

Party A	Party B (Routing Point/CDN)	Party C
Make incoming call to information service		
EventDialing ConnID 1 ThisDN A ThisDNRole Origination OtherDN B OtherDNRole Destination	EventQueued ConnID 1 ThisDN B ThisQueue B ThisDNRole Destination OtherDN A OtherDNRole Origination EventRouteRequest ConnID 1 ThisDN B ThisQueue B ThisDNRole Destination OtherDN A OtherDNRole Origination	
	Route call to C ^a (TRouteCall)	
	EventRouteUsed ConnID 1 ThisDN B ThisDNRole Destination OtherDN A OtherDNRole Origination ThirdPartyDN C *OPT ThirdPartyDNRole Destination *OPT EventDiverted ConnID 1 ThisDN B ThisQueue B ThisDNRole Destination OtherDN A OtherDNRole Origination ThirdPartyDN C *OPT ThirdPartyDNRole Destination *OPT	
		EventRinging ConnID 1 ThisDN C ThisQueue B ThisDNRole Destination OtherDN A OtherDNRole Origination CallState OK
		Answer (TAnswerCall)
EventEstablished		EventEstablished

ConnID 1 ThisDN A ThisDNRole Origination OtherDN C OtherDNRole Destination		ConnID 1 ThisDN C ThisDNRole Destination OtherDN A OtherDNRole Origination
Conversation		

a. RouteCall to C (TRouteCall()) might be missing.

Abnormal call flow

Interruption point	Party A	Party B	Party C
* and **	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventAbandoned ConnID 1 ThisDN B OtherDN A CallState OK	
***	EventReleased ConnID 1 ThisDN A OtherDN C CallState OK		
****	EventReleased ConnID 1 ThisDN A OtherDN C CallState OK		EventAbandoned ConnID 1 ThisDN C OtherDN A CallState OK

Connection-establishing phase for internal/inbound call with routing

The following table describes the connection-establishing phase for an internal/inbound call with routing.

Party A	Party B (Routing Point/CDN)	Party C
Make incoming call to information service		
EventDialing ConnID 1 ThisDN A ThisDNRole Origination OtherDN B *DIAL OtherDNRole Destination *DIAL	EventRouteRequest ConnID 1 ThisDN B ThisDNRole Destination OtherDN A OtherDNRole Origination	

	Route call to C ^a (TRouteCall)	
	EventRouteUsed ConnID 1 ThisDN B ThisDNRole Destination OtherDN A OtherDNRole Origination ThirdPartyDN C ^b ThirdPartyDNRole Destination *OPT CallState OK/Redirected ^c	
		EventRinging ConnID 1 ThisDN C ThisDNRole Destination OtherDN A OtherDNRole Origination CallState OK
		Answer (TAnswerCall)
EventEstablished ConnID 1 ThisDN A ThisDNRole Origination OtherDN C OtherDNRole Destination		EventEstablished ConnID 1 ThisDN C ThisDNRole Destination OtherDN A OtherDNRole Origination
Conversation		

a. Not present if a call has been routed by default; that is, a switch did not receive any routing instruction from a computer domain within a timeout configured on the switch side (scripted or otherwise) and therefore processed the call using switch logic.

b. Content of **ThirdPartyDN** depends on the call scenario:

- If information about the destination is available at the moment EventRouteUsed is generated, this attribute is mandatory; a DN where the call has been delivered must be reported.
- If the information is not available, but the call has been routed through Voice Microservices, this attribute is mandatory; a DN where the call has been sent must be reported.
- If a call has been routed to a default destination or routed by another application, this attribute is optional (depends on switch capabilities).

c. **CallState** has a value of Redirected (22) if a call has been routed by a switch. For some switches, the attribute **Callstate** might not be present.

Abnormal call flow

Interruption point	Party A	Party B	Party C
*	EventReleased ConnID 1 ThisDN A OtherDN B	EventAbandoned ConnID 1 ThisDN B OtherDN A	

	CallState OK	CallState OK	
**	EventReleased ConnID 1 ThisDN A OtherDN C CallState OK	EventAbandoned ^a ConnID 1 ThisDN B OtherDN A CallState OK	
***	EventReleased ConnID 1 ThisDN A OtherDN C CallState OK		
****	EventReleased ConnID 1 ThisDN A OtherDN C CallState OK		EventAbandoned ConnID 1 ThisDN C OtherDN A CallState OK

a. In this case, EventError must be sent after EventAbandoned to make the ReferenceID available.

Connection-establishing phase for an internal/inbound call with routing outbound

The following table describes the connection-establishing phase for an internal/inbound call with routing outbound.

Party A	Party B (Routing Point)	Party C
Incoming call		
EventDialing ConnID 1 ThisDN A ThisDNRole Origination OtherDN B ^{*DIAL} OtherDNRole Destination ^{*DIAL}	EventRouteRequest ConnID 1 ThisDN B ThisDNRole Destination OtherDN A OtherDNRole Origination	
	Route call to C ^a (TRouteCall)	
EventNetworkReached ConnID 1 ThisDN A ThisDNRole Origination OtherDN C ^{*DIAL} OtherDNRole Destination ^{*DIAL}	EventRouteUsed ConnID 1 ThisDN B ThisDNRole Destination OtherDN A OtherDNRole Origination ThirdPartyDN C ^b ThirdPartyDNRole Destination ^{*OPT} CallState OK/Redirected ^c	EventRinging ConnID 1 ThisDN C ThisDNRole Destination OtherDN A OtherDNRole Origination CallState OK

		Answer (TAnswerCall)
EventEstablished ConnID 1 ThisDN A ThisDNRole Origination OtherDN C OtherDNRole Destination		EventEstablished ConnID 1 ThisDN C ThisDNRole Destination OtherDN A OtherDNRole Origination
Conversation		

a. Not present if a call has been routed by default; that is, a switch did not receive any routing instruction from a computer domain within a timeout configured on the switch side (scripted or otherwise) and therefore processed the call using switch logic.

b. Content of **ThirdPartyDN** depends on the call scenario:

- If information about the destination is available at the moment EventRouteUsed is generated, this attribute is mandatory; a DN where the call has been delivered must be reported.
- If the information is not available, but the call has been routed through Voice Microservices, this attribute is mandatory; a DN where the call has been sent must be reported.
- If a call has been routed to a default destination or routed by another application, this attribute is optional (depends on switch capabilities).

c. **CallState** has a value of Redirected (22) if a call has been routed by a switch. For some switches, the attribute **CallState** might not be present.

Abnormal call flow

Interruption point	Party A	Party B	Party C
*	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventAbandoned ConnID 1 ThisDN B OtherDN A CallState OK	
**	EventReleased ConnID 1 ThisDN A OtherDN C CallState OK		EventAbandoned ConnID 1 ThisDN C OtherDN A CallState OK

Connection-establishing phase for an outbound call

The following table describes the connection-establishing phase for an outbound call.

Party A	Party B
Make outside call (TMakeCall)	
EventDialing	

ConnID 1 ThisDN A ThisDNRole Origination OtherDN B *DIAL OtherDNRole Destination *DIAL	
EventNetworkReached ^a ConnID 1 ThisDN A ThisDNRole Origination OtherDN B *DIAL OtherDNRole Destination *DIAL	
	Answer
EventEstablished ConnID 1 ThisDN A ThisDNRole Origination OtherDN B *OPT OtherDNRole Destination *OPT	
Conversation	

a. When a switch does not report network reached, Voice Microservices simulate EventNetworkReached right before distributing EventEstablished.

Abnormal call flow

Interruption point	Party A
*	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK
**	EventDestinationBusy ConnID 1 ThisDN A OtherDN B CallState ^a
***	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK

a. CallState might have values that clarify the reason for the destination being busy, for instance CallStateSitInvalidNum.

Connection-establishing phase while on hold (internal/outbound call)

The following table describes the connection-establishing phase for an internal/outbound call while on hold.

Party A	Party B
Call to B	
EventDialing ConnID 1 ThisDN A ThisDNRole Origination OtherDN B OtherDNRole Destination CallState OK	EventRinging ConnID 1 ThisDN B ThisDNRole Destination OtherDN A OtherDNRole Origination CallState OK
Hold	
EventHeld ConnID 1 ThisDN A OtherDN B	
	Answer
EventEstablished ConnID 1 ThisDN A OtherDN B	EventEstablished ConnID 1 ThisDN B OtherDN A
Retrieve	
EventRetrieved ConnID 1 ThisDN A OtherDN B CallState OK	

Releasing calls

Contents

- [1 Release phase](#)
- [2 Release from conference phase](#)
- [3 Delete from conference phase](#)

Learn about the standard processes by which calls are released.

Related documentation:

-
-
-

RSS:

- [For private edition](#)

For simplicity, the examples on this page use abbreviated attribute values. For example, ConnID **1**, which – in actual events – displays as ConnID>@metainformation>.

The following comments and abbreviations are used in the call models:

- OPT—Optional.
- DIAL—Might be a dialed number or is not present if Voice Microservices have no information about the other party.

Release phase

The following table describes the release phase.

Party A	Party B
Conversation	
Release (TReleaseCall)	
EventReleased ConnID 1 ThisDN A OtherDN B *OPT CallState OK	EventReleased ConnID 1 ThisDN B OtherDN A *OPT CallState OK

Release from conference phase

The following table describes the release from conference phase.

Party A	Party B	Party C

Conference		
	Release (TReleaseCall)	
EventPartyDeleted ConnID 1 ThisDN A OtherDN B OtherDNRole DeletedParty ThirdPartyDN B ThirdPartyDNRole DeletedBy CallState OK/Conferenced ^a	EventReleased ConnID 1 ThisDN B CallState OK	EventPartyDeleted ConnID 1 ThisDN C OtherDN B OtherDNRole DeletedParty ThirdPartyDN B ThirdPartyDNRole DeletedBy CallState OK/Conferenced ^a
Conversation		

a. If more than two parties remain in the conference call, **CallState** has a value of Conferenced; otherwise, **CallState** has a value of OK.

Delete from conference phase

The following table describes the release phase.

Party A	Party B	Party C
Conference		
Delete B (TDeleteFromConference)		
EventPartyDeleted ConnID 1 ThisDN A OtherDN B OtherDNRole DeletedParty ThirdPartyDN A ThirdPartyDNRole DeletedBy CallState OK/Conferenced ^a	EventReleased ConnID 1 ThisDN B CallState OK	EventPartyDeleted ConnID 1 ThisDN C OtherDN B OtherDNRole DeletedParty ThirdPartyDN A ThirdPartyDNRole DeletedBy CallState OK/Conferenced ^a
Conversation		

a. If more than two parties remain in the conference call, **CallState** has a value of Conferenced; otherwise, **CallState** has a value of OK.

Holding, transferring, and conferencing

Contents

- 1 Hold/Retrieve function, consulted party answers
 - 1.1 Abnormal call flow
- 2 Hold/Retrieve function, consulted party does not answer
 - 2.1 Abnormal call flow
- 3 Single-step transfer
 - 3.1 Abnormal call flow
- 4 Single-step transfer (outbound)
 - 4.1 Abnormal call flow
- 5 Mute transfer
 - 5.1 Abnormal call flow
- 6 Two-step transfer: complete after consulted party answers
 - 6.1 Abnormal call flow
- 7 Two-step transfer: complete before consulted party answers (blind)
 - 7.1 Abnormal call flow
- 8 Two-step transfer to ACD
 - 8.1 Abnormal call flow
- 9 Two-step transfer to a Routing Point
 - 9.1 Abnormal call flow
- 10 Single-step conference
- 11 Conference
 - 11.1 Abnormal call flow
- 12 Blind conference (complete before consulted party answers)
 - 12.1 Abnormal call flow

Learn about the functions and events related to placing calls on hold, transferring calls, and creating conference calls.

Related documentation:

-
-
-

RSS:

- [For private edition](#)

For simplicity, the examples on this page use abbreviated attribute values. For example, ConnID **1**, which – in actual events – displays as ConnID>@metainformation>.

The following comments and abbreviations are used in the call models:

- OPT—Optional.
- DIAL—Might be a dialed number or is not present if Voice Microservices have no information about the other party.

Hold/Retrieve function, consulted party answers

Party A	Party B	Party C
Call-establishing phase (ConnID 1)		
	Hold (THoldCall)	
	EventHeld ConnID 1 ThisDN B OtherDN A	
	Make call to C (Consultation) (TMakeCall)	
Call-establishing phase (ConnID 2)		
Release phase (ConnID 2)		
	Retrieve call from A (TRetrieveCall)	
	EventRetrieved ^a	

	ConnID 1 ThisDN B OtherDN A CallState OK	
Release phase (ConnID 1)		

a. With EventRetrieved, the values for attributes **ThisDNRole** and **ThisQueue** are the same as those for the attributes of the same names, if any, in the events preceding EventRetrieved (EventEstablished and EvenRinging). For non-ACD calls, however, **ThisQueue** is not reported.

Abnormal call flow

Interruption point	Party A	Party B
*	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventReleased ConnID 1 ThisDN B OtherDN A CallState OK
**	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventReleased ConnID 1 ThisDN B OtherDN A CallState OK
***	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventReleased ConnID 1 ThisDN B OtherDN A CallState OK

Hold/Retrieve function, consulted party does not answer

Party A	Party B	Party C
Call-establishing phase (ConnID 1)		
	Hold (THoldCall)	
	EventHeld ConnID 1 ThisDN B OtherDN A	
Unsuccessful internal call (party does not answer) (ConnID 2)		
	Retrieve call from A (TRetrieveCall)	

	EventRetrieved^a ConnID 1 ThisDN B OtherDN A CallState OK	
Release phase (ConnID 1)		

a. With EventRetrieved, the values for attributes **ThisDNRole** and **ThisQueue** are the same as those for the attributes of the same names, if any, in the events preceding EventRetrieved (EventEstablished and EvenRinging). For non-ACD calls, however, **ThisQueue** is not reported.

Abnormal call flow

Interruption point	Party A	Party B
*	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventReleased ConnID 1 ThisDN B OtherDN A CallState OK
**	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventReleased ConnID 1 ThisDN B OtherDN A CallState OK

Single-step transfer

Party A	Party B	Party C
Call-establishing phase (ConnID 1)		
Single-step transfer to C (TSingleStepTransfer)		
EventPartyChanged ConnID 1 PreviousConnID 1 ThisDN A OtherDN C ThirdPartyDN B ThirdPartyDNRole TransferredBy CallState Transferred	EventReleased ConnID 1 ThisDN B ThirdPartyDN C OtherDN A CallState Transferred Cause 1stepTransfer	EventRinging ConnID 1 ThisDN C OtherDN A ThirdPartyDN B ThirdPartyDNRole TransferredBy CallState Transferred
		Answer (TAnswerCall)

		EventEstablished ConnID 1 ThisDN C OtherDN A
--	--	--

Abnormal call flow

Interruption point	Party A	Party B	Party C
*	EventReleased ConnID 1 ThisDN A OtherDN C CallState OK		
**	EventReleased ConnID 1 ThisDN A OtherDN C CallState OK		EventAbandoned ConnID 1 ThisDN C OtherDN A CallState OK

Single-step transfer (outbound)

Party A	Party B	Party C
Call-establishing phase (ConnID 1)		
	Single-step transfer to C (TSingleStepTransfer)	
EventPartyChanged ConnID 1 PreviousConnID 1 ThisDN A OtherDN C ThirdPartyDN B ThirdPartyDNRole TransferredBy CallState Transferred EventNetworkReached ConnID 1 ThisDN A OtherDN C *DIAL OtherDNRole Destination *DIAL	EventReleased ConnID 1 ThisDN B ThirdPartyDN C OtherDN A CallState Transferred Cause 1stepTransfer	EventRinging ConnID 1 ThisDN C OtherDN A ThirdPartyDN B ThirdPartyDNRole TransferredBy CallState Transferred
		Answer (TAnswerCall)
		EventEstablished ConnID 1 ThisDN C OtherDN A

Abnormal call flow

Interruption point	Party A	Party B	Party C
*	EventReleased ConnID 1 ThisDN A OtherDN C CallState OK		
**	EventReleased ConnID 1 ThisDN A OtherDN C CallState OK		EventAbandoned ConnID 1 ThisDN C OtherDN A CallState OK

Mute transfer

Party A	Party B	Party C
Call-establishing phase (ConnID 1)		
	Mute transfer to C (TMuteTransfer*)	
	EventHeld ConnID 1 ThisDN B OtherDN A	
	EventDialing ConnID 2 ThisDN B ThisDNRole Origination OtherDN C OtherDNRole Destination	EventRinging ConnID 2 ThisDN C ThisDNRole Destination OtherDN B OtherDNRole Origination CallState OK
EventPartyChanged ConnID 1 PreviousConnID 1 ThisDN A OtherDN C ThirdPartyDN B ThirdPartyDNRole TransferredBy CallState Transferred	EventReleased ConnID 1 ThisDN B OtherDN A CallState Transferred EventReleased ConnID 2 ThisDN B ThisDNRole Origination OtherDN C OtherDNRole Destination CallState Transferred	EventPartyChanged ConnID 1 PreviousConnID 2 ThisDN C OtherDN A ThirdPartyDN B ThirdPartyDNRole TransferredBy CallState Transferred

		Answer (TAnswerCall)
		EventEstablished
		ConnID 1 ThisDN C OtherDN A
Release phase (ConnID 1)		

Abnormal call flow

Interruption point	Party A	Party B	Party C
*	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventReleased ConnID 1 ThisDN B OtherDN A CallState OK	
**	EventReleased ConnID 1 ThisDN A OtherDN C CallState OK	EventReleased ConnID 1 ThisDN B OtherDN A CallState OK EventReleased ConnID 2 ThisDN B OtherDN C CallState OK	EventAbandoned ConnID 2 ThisDN C OtherDN B CallState OK
***	EventReleased ConnID 1 ThisDN B OtherDN C CallState OK		EventAbandoned ConnID 1 ThisDN C OtherDN B CallState OK

Two-step transfer: complete after consulted party answers

Party A	Party B	Party C
Call-establishing phase (ConnID 1)		
	Hold (TInitiateTransfer*)	
	EventHeld ConnID 1 ThisDN B OtherDN A	
	Consultation call to C (TInitiateTransfer continues)	

Call-establishing phase (ConnID 2)		
	Transfer held call to C (TCompleteTransfer)	
EventPartyChanged ConnID 1 PreviousConnID 1 ThisDN A OtherDN C ThirdPartyDN B ThirdPartyDNRole TransferredBy CallState Transferred	EventReleased ConnID 1 ThisDN B OtherDN A CallState Transferred EventReleased ConnID 2 ThisDN B OtherDN C CallState Transferred	EventPartyChanged ConnID 1 PreviousConnID 2 ThisDN C OtherDN A ThirdPartyDN B ThirdPartyDNRole TransferredBy CallState Transferred
Release phase (ConnID 1)		

Abnormal call flow

Interruption point	Party A	Party B	Party C
*	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventReleased ConnID 1 ThisDN B OtherDN A CallState OK	
**	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventReleased ConnID 1 ThisDN B OtherDN A CallState OK EventReleased ConnID 2 ThisDN B OtherDN C CallState OK	EventAbandoned ConnID 2 ThisDN C OtherDN B CallState OK

Two-step transfer: complete before consulted party answers (blind)

Party A	Party B	Party C
Call-establishing phase (ConnID 1)		
	Hold (TInitiateTransfer)	
	EventHeld	

	ConnID 1 ThisDN B OtherDN A	
	Consultation call to C (TInitiateTransfer continues)	
	EventDialing ConnID 2 ThisDN B ThisDNRole Origination OtherDN C *DIAL OtherDNRole Destination *DIAL	EventRinging ConnID 2 ThisDN C ThisDNRole Destination OtherDN B OtherDNRole Origination CallState OK
	Transfer held call to C (TCompleteTransfer)	
EventPartyChanged ConnID 1 PreviousConnID 1 ThisDN A OtherDN C ThirdPartyDN B ThirdPartyDNRole TransferredBy CallState Transferred	EventReleased ConnID 1 ThisDN B OtherDN A CallState Transferred EventReleased ConnID 2 ThisDN B OtherDN C CallState Transferred	EventPartyChanged ConnID 1 PreviousConnID 2 ThisDN C OtherDN A ThirdPartyDN B ThirdPartyDNRole TransferredBy CallState Transferred
		Answer (TAnswerCall)
		EventEstablished ConnID 1 ThisDN C OtherDN A
Release phase (ConnID 1)		

If a call appears on the terminating party after transfer completion, the **ConnID** field of EventRinging is equal to the connection ID of the original call (ConnID 1), and EventPartyChanged is not generated.

Abnormal call flow

Interruption point	Party A	Party B	Party C
*	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventReleased ConnID 1 ThisDN B OtherDN A CallState OK	
**	EventReleased	EventReleased	EventAbandoned

	ConnID 1 ThisDN A OtherDN B CallState OK	ConnID 1 ThisDN B OtherDN A CallState OK EventReleased ConnID 2 ThisDN B OtherDN C CallState OK	ConnID 2 ThisDN C OtherDN B CallState OK
***	EventReleased ConnID 1 ThisDN A OtherDN C CallState OK		EventAbandoned ConnID 1 ThisDN C OtherDN A CallState OK

Two-step transfer to ACD

Two-step transfer to ACD means that a call is waiting in a queue, and the transfer completed before any ACD agent is available to receive the call.

The following table describes a two-step transfer to ACD.

Party A	Party B	Party C (ACD)	Party D
Call-establishing phase (ConnID 1)			
	Hold (TInitiateTransfer)		
	EventHeld ConnID 1 ThisDN B OtherDN A		
	Consultation call to C (TInitiateTransfer continues)		
	EventDialing ConnID 2 ThisDN B OtherDN C *DIAL	EventQueued ConnID 2 ThisDN C ThisQueue C OtherDN B	
	Transfer held call to C (TCompleteTransfer)		
EventPartyChanged ConnID 1 PreviousConnID 1 ThisDN A OtherDN C	EventReleased ConnID 2 ThisDN B ThisDNRole Origination OtherDN C	EventPartyChanged ConnID 1 PreviousConnID 2 ThisDN C ThisQueue C	

ThirdPartyDN B ThirdPartyDNRole TransferredBy CallState Transferred	OtherDNRole Destination CallState Transferred EventReleased ConnID 1 ThisDN B OtherDN A CallState Transferred	OtherDN A ThirdPartyDN B ThirdPartyDNRole TransferredBy CallState Transferred	
		Diverts call to D	
		EventDiverted ConnID 1 ThisDN C OtherDN A ThirdPartyDN C *OPT ThirdPartyDNRole Destination *OPT	EventRinging ConnID 1 ThisDN D ThisQueue C OtherDN A CallState OK
			Answer (TAnswerCall)
			EventEstablished ConnID 1 ThisDN D ThisQueue C OtherDN A CallState OK
Release phase (ConnID 1)			

If a call transfer is completed before it is put in an ACD queue, an EventPartyChanged is not generated.

Abnormal call flow

Interruption point	Party A	Party B	Party C	Party D
*	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventReleased ConnID 1 ThisDN B OtherDN A CallState OK		
**	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventReleased ConnID 1 ThisDN B OtherDN A CallState OK EventReleased ConnID 2 ThisDN B OtherDN C CallState OK	EventAbandoned ConnID 2 ThisDN C OtherDN B CallState OK	
***	EventReleased			EventAbandoned

	ConnID 1 ThisDN A OtherDN D CallState OK			ConnID 1 ThisDN D OtherDN A CallState OK
--	---	--	--	---

Two-step transfer to a Routing Point

Party A	Party B	Party C (ACD)	Party D
Call-establishing phase (ConnID 1)			
	Hold (TInitiateTransfer)		
	EventHeld ConnID 1 ThisDN B OtherDN A		
	Consultation call to C (TInitiateTransfer continues)		
	EventDialing ConnID 2 ThisDN B ThisDNRole Origination OtherDN C ^{*DIAL} OtherDNRole Destination CallType Consult	EventRouteRequest ConnID 2 ThisDN C ThisDNRole Destination OtherDN B OtherDNRole Origination	
	Transfer held call to C (TCompleteTransfer)		
EventPartyChanged ConnID 1 PreviousConnID 1 ThisDN A ThisDNRole Origination ^a OtherDN C ThirdPartyDN B ThirdPartyDNRole TransferredBy CallState Transferred	EventReleased ConnID 2 ThisDN B ThisDNRole Origination OtherDN C OtherDNRole Destination CallState Transferred EventReleased ConnID 1 ThisDN B ThisDNRole Destination OtherDN A CallState Transferred	EventPartyChanged ConnID 1 PreviousConnID 2 ThisDN C OtherDN A ThirdPartyDN B ThirdPartyDNRole TransferredBy CallState Transferred	
		Diverts call to D	
		EventRouteUsed ConnID 1 ThisDN C OtherDN A	EventRinging ConnID 1 ThisDN D OtherDN A CallState OK

		ThirdPartyDN D *OPT	
			Answer (TAnswerCall)
			EventEstablished ConnID 1 ThisDN D OtherDN A
Release phase (ConnID 1)			

a. **ThisDNRole** must be Destination if party B is the call originator.

Abnormal call flow

Interruption point	Party A	Party B	Party C	Party D
*	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventReleased ConnID 1 ThisDN B OtherDN A CallState OK		
**	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventReleased ConnID 1 ThisDN B OtherDN A CallState OK EventReleased ConnID 2 ThisDN B OtherDN C CallState OK	EventAbandoned ConnID 2 ThisDN C OtherDN B CallState OK	
***	EventReleased ConnID 1 ThisDN A OtherDN C CallState OK		EventAbandoned ConnID 1 ThisDN C OtherDN A CallState OK	
****	EventReleased ConnID 1 ThisDN A OtherDN D CallState OK			EventAbandoned ConnID 1 ThisDN D OtherDN A CallState OK

Single-step conference

Party A	Party B	Party C

Call-establishing phase (ConnID 1)		
	TSingleStepConference	
EventPartyAdded ConnID 1 ThisDN A OtherDN C ThirdPartyDN B ^a	EventPartyAdded ConnID 1 ThisDN B OtherDN C ThirdPartyDN B ^a	EventRinging ConnID 1 ThisDN C ThisDNRole ConferenceMember CallState OK
		EventEstablished ConnID 1 ThisDN C ThisDNRole ConferenceMember CallState Conferenced
Release from conference phase		
Release phase (ConnID 1)		

a. **ThirdPartyDN** has a value of C if Party C initiates the request for a conference.

Conference

Party A	Party B	Party C
Call-establishing phase (ConnID 1)		
	Hold	
	EventHeld ConnID 1 ThisDN B ThisDNRole Previous Role of DN OtherDN A OtherDNRole Previous Role of DN	
	Consultation call to C	
Call-establishing phase (ConnID 2)		
	Conference	
	EventReleased ConnID 2 ThisDN B OtherDN C CallState Conferenced EventRetrieved ^a ConnID 1 ThisDN B OtherDN A CallState Conferenced	

EventPartyAdded ConnID 1 ThisDN A OtherDN C ThirdPartyDN B ThirdPartyDNRole AddedBy CallState Conferenced	EventPartyAdded ConnID 1 ThisDN B OtherDN ^b C OtherDNRole NewParty ThirdPartyDN B ThirdPartyDNRole AddedBy CallState Conferenced	EventPartyChanged ConnID 1 PreviousConnID 2 ThisDN C ThirdPartyDN B ThirdPartyDNRole ConferencedBy CallState Conferenced
Release from conference phase		
Release phase (ConnID 1)		

- a. With EventRetrieved, the values for attributes **ThisDNRole** and **ThisQueue** are the same as those for the attributes of the same names, if any, in the events preceding EventRetrieved (EventEstablished and EventRinging). For non-ACD calls, however, **ThisQueue** is not reported.
- b. If only one party is added (as in the case of a simple conference call), the corresponding telephony object is specified in **OtherDN**. If more than one party is added, then the corresponding telephony objects are specified in **Extensions**.

Abnormal call flow

Interruption point	Party A	Party B
*	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventReleased ConnID 1 ThisDN B OtherDN A CallState OK

Blind conference (complete before consulted party answers)

Party A	Party B	Party C
Call-establishing phase (ConnID 1)		
	Hold	
	EventHeld ConnID 1 ThisDN B OtherDN A	
	Consultation call to C	
	EventDialing ConnID 2 ThisDN B ThisDNRole Origination OtherDN C ^{*DIAL} OtherDNRole Destination ^{*DIAL} CallType Consult	EventRinging ConnID 2 ThisDN C ThisDNRole Destination OtherDN B OtherDNRole Origination CallType Consult

	Conference	
EventPartyAdded ConnID 1 ThisDN A OtherDN C ThirdPartyDN B ThirdPartyDNRole AddedBy CallState Conferenced	EventReleased ConnID 2 ThisDN B OtherDN C CallState Conferenced EventRetrieved ^a ConnID 1 ThisDN B OtherDN A CallState Conferenced EventPartyAdded ConnID 1 ThisDN B OtherDN C ThirdPartyDN B ThirdPartyDNRole AddedBy CallState Conferenced	EventPartyChanged ConnID 1 PreviousConnID 2 ThisDN C ThirdPartyDN B ThirdPartyDNRole ConferencedBy CallState Conferenced
		Answer (TAnswerCall)
		EventEstablished ConnID 1 ThisDN C CallState Conferenced
Release from conference phase		
Release phase (ConnID 1)		

a. With EventRetrieved, the values for attributes **ThisDNRole** and **ThisQueue** are the same as those for the attributes of the same names, if any, in the events preceding EventRetrieved (EventEstablished and EventRinging). For non-ACD calls, however, **ThisQueue** is not reported.

If a call appears on the terminating party after completion of conference, the **ConnID** field of EventRinging is equal to the connection ID of the original call (ConnID 1), and EventPartyChanged is not generated.

Abnormal call flow

Interruption point	Party A	Party B	Party C
*	EventReleased ConnID 1 ThisDN A OtherDN B CallState OK	EventReleased ConnID 1 ThisDN B OtherDN A CallState OK	
**	EventReleased ConnID 1 ThisDN A OtherDN B *DIAL CallState OK	EventPartyDeleted ConnID 1 ThisDN B OtherDN A OtherDNRole DeletedParty ThirdPartyDN A ThirdPartyDNRole DeletedBy	

		CallState OK	
--	--	---------------------	--

Predictive dialing

Contents

- [1 Predictive call](#)
 - [1.1 Abnormal call flow](#)
- [2 Predictive call with routing](#)
 - [2.1 Abnormal call flow](#)
- [3 Predictive call \(connected to a device specified in Extensions\)](#)
 - [3.1 Abnormal call flow](#)

Learn about call models and flows for predictive dialing.

Related documentation:

-
-
-

RSS:

- [For private edition](#)

For simplicity, the examples on this page use abbreviated attribute values. For example, ConnID **1**, which – in actual events – displays as ConnID>@metainformation>.

The following comments and abbreviations are used in the call models:

- OPT—Optional.
- DIAL—Might be a dialed number or is not present if Voice Microservices have no information about the other party.

Predictive call

The following table describes a predictive call.

Party A	Party B (ACD Group)	Party C
	Make predictive call (TMakePredictiveCall)	
	EventDialing ConnID 1 ThisDN B ThisQueue B ThisDNRole Origination OtherDN C *DIAL OtherDNRole Destination	
		Answer
	EventQueued ConnID 1 ThisDN B ThisQueue B ThisDNRole Origination CallState OK / AnsweringMachineDetected ^a	

	EventDiverted ConnID 1 ThisDN B ThisQueue B ThisDNRole Origination OtherDN C OtherDNRole Destination ThirdPartyDN A *OPT ThirdPartyDNRole Origination *OPT	
EventRinging ConnID 1 ThisDN A ThisDNRole Origination OtherDN C OtherDNRole Destination CallState OK		
Answer (TAnswerCall)		
EventEstablished ConnID 1 ThisDN A ThisDNRole Origination OtherDN C OtherDNRole Destination		
Release Phase (ConnID 1)		

a. If the switch reports that a call is connected to an answering machine, Voice Microservices also attach a key-value pair AnswerClass=AM to the call's UserData.

Abnormal call flow

Interruption point	Party A	Party B	Party C
*		EventReleased ConnID 1 ThisDN B OtherDN C CallState ^a	
**		EventAbandoned ConnID 1 ThisDN B OtherDN C CallState OK	
***	EventAbandoned ConnID 1 ThisDN A OtherDN C CallState OK		

a. **CallState** in this case can be any of the following:

- CallStateGeneralError
- CallStateSystemError
- CallStateBusy
- CallStateNoAnswer
- CallStateAnsweringMachineDetected
- CallStateFaxDetected
- CallStateAllTrunksBusy
- CallStateQueueFull
- CallStateDropped
- CallStateSitDetected
- CallStateSitInvalidnum
- CallStateSitVacant
- CallStateSitIntercept
- CallStateSitUnknown
- CallStateSitNocircuit
- CallStateSitReorder

Predictive call with routing

The following table describes a predictive call with routing.

Party A	Party B (ACD Group)	Party C
	Make Predictive Call (TMakePredictiveCall)	
	EventDialing ConnID 1 ThisDN B ThisQueue B ThisDNRole Origination OtherDN C *DIAL OtherDNRole Destination	
		Answer
	EventQueued ConnID 1 ThisDN B ThisQueue B ThisDNRole Origination CallState OK / FaxDetected / AnsweringMachineDetected ^a	
	EventRouteRequest	

	ConnID 1 ThisDN B ThisQueue B ThisDNRole Origination OtherDN C OtherDNRole Destination	
	Route call to A (TRouteCall)	
	EventRouteUsed ConnID 1 ThisDN B ThisDNRole Origination OtherDN C OtherDNRole Destination ThirdPartyDN A *OPT ThirdPartyDNRole Origination *OPT EventDiverted ConnID 1 ThisDN B ThisQueue B ThisDNRole Origination OtherDN C OtherDNRole Destination ThirdPartyDN A *OPT ThirdPartyDNRole Origination *OPT	
EventRinging ConnID 1 ThisDN A ThisDNRole Origination OtherDN C OtherDNRole Destination CallState OK		
Answer (TAnswerCall)		
EventEstablished ConnID 1 ThisDN A ThisDNRole Origination OtherDN C OtherDNRole Destination		
Release Phase (ConnID 1)		

a. If the switch reports that a call is connected to an answering machine, Voice Microservices also attach a key-value pair AnswerClass=AM to the call's UserData.

Abnormal call flow

Interruption point	Party A	Party B	Party C
*		EventReleased ConnID 1 ThisDN B OtherDN C	

		CallState ^a	
** and ***		EventAbandoned ConnID 1 ThisDN B OtherDN C CallState OK	
****	EventAbandoned ConnID 1 ThisDN A OtherDN C CallState OK		

a. **CallState** in this case can be any of the following:

- CallStateGeneralError
- CallStateSystemError
- CallStateBusy
- CallStateNoAnswer
- CallStateAnsweringMachineDetected
- CallStateFaxDetected
- CallStateAllTrunksBusy
- CallStateQueueFull
- CallStateDropped
- CallStateSitDetected
- CallStateSitInvalidnum
- CallStateSitVacant
- CallStateSitIntercept
- CallStateSitUnknown
- CallStateSitNocircuit
- CallStateSitReorder

Predictive call (connected to a device specified in Extensions)

The following table describes a predictive call (connected to a device specified in extensions).

Party A	Party B (ACD Group Specified in the Extensions of TMakePredictiveCall)	Party C (Routing Point or ACD Group)	Party D
---------	---	--	---------

		Make predictive call (TMakePredictiveCall)	
		EventDialing ConnID 1 ThisDN C ThisQueue C ThisDNRole Origination OtherDN D *DIAL OtherDNRole Destination	
			Answer
		EventQueued ConnID 1 ThisDN C ThisQueue C ThisDNRole Origination CallState OK/ AnsweringMachine- Detected	
		EventDiverted ConnID 1 ThisDN C ThisQueue C ThisDNRole Origination OtherDN D OtherDNRole Destination ThirdPartyDN B ThirdPartyDNRole Origination	
	EventQueued ConnID 1 This DN B ThisQueue B ThisDNRole Origination OtherDN D OtherDNRole Destination		
	EventDiverted ConnID 1 ThisDN B ThisQueue B ThisDNRole Origination OtherDN D OtherDNRole Destination ThirdPartyDN A *OPT ThirdPartyDNRole Origination *OPT		
EventRinging ConnID 1 ThisDN A ThisDNRole Origination OtherDN D OtherDNRole Destination CallState OK			
Answer (TAnswerCall)			

EventEstablished ConnID 1 ThisDN A ThisDNRole Origination OtherDN D OtherDNRole Destination			
Release Phase (ConnID 1)			

Abnormal call flow

Interruption point	Party A	Party B	Party C	Party D
*			EventReleased ConnID 1 ThisDN C OtherDN D CallState ^a	
**		EventAbandoned ConnID 1 ThisDN B OtherDN D CallState OK		
***	EventAbandoned ConnID 1 ThisDN A OtherDN D CallState OK			

a. **CallState** in this case can be any of the following:

- CallStateGeneralError
- CallStateSystemError
- CallStateBusy
- CallStateNoAnswer
- CallStateAnsweringMachineDetected
- CallStateFaxDetected
- CallStateAllTrunksBusy
- CallStateQueueFull
- CallStateDropped
- CallStateSitDetected
- CallStateSitInvalidnum
- CallStateSitVacant

- CallStateSitIntercept
- CallStateSitUnknown
- CallStateSitNocircuit
- CallStateSitReorder

Monitoring calls

Contents

- 1 Service observing on agent
 - 1.1 Agent releases first
 - 1.2 External party releases first
 - 1.3 Observer releases first
 - 1.4 Abnormal call flow
- 2 Service observing for agent-initiated call
- 3 Service observing on queue
 - 3.1 Abnormal call flow

Learn about call monitoring models and flows.

Related documentation:

-
-
-

RSS:

- [For private edition](#)

For simplicity, the examples on this page use abbreviated attribute values. For example, ConnID **1**, which – in actual events – displays as ConnID>@metainformation>.

The following comments and abbreviations are used in the call models:

- OPT—Optional.
- DIAL—Might be a dialed number or is not present if Voice Microservices have no information about the other party.

Service observing on agent

The following tables describe service observing on an agent.

Party A (External)	Party B	Party C (Service Observer)
Inbound call		
	EventRinging ConnID 1 ThisDN B ThisDNRole Destination OtherDN A OtherDNRole Origination CallState OK	
	Answer (TAnswerCall)	
	EventEstablished ConnID 1 ThisDN B ThisDNRole Destination OtherDN A OtherDNRole Origination CallState OK	EventRinging ConnID 1 ThisDN C ThisDNRole Observer OtherDN A ^a OtherDNRole Origination ^b

		CallState Bridged
	EventPartyAdded ConnID 1 ThisDN B ThisDNRole Destination OtherDN C OtherDNRole Observer CallState Bridged	EventEstablished ConnID 1 ThisDN C ThisDNRole Observer OtherDN A OtherDNRole Origination CallState Bridged
Conference		
Release phase (see descriptions below)		

- a. For some switches, Voice Microservices use the party that initialized the Service Observer instead of Party A.
- b. For some switches, Voice Microservices use the role of the party that initialized the Service Observer instead of the role of Party A.

Agent releases first

Party A (External)	Party B	Party C (Service Observer)
	Release (TReleaseCall)	
	EventReleased ConnID 1 ThisDN B ThisDNRole Destination CallState OK	EventPartyDeleted ConnID 1 ThisDN C ThisDNRole Observer OtherDN B OtherDNRole DeletedParty CallState OK
		EventReleased ConnID 1 ThisDN C ThisDNRole Observer OtherDN A CallState OK

External party releases first

Party A (External)	Party B	Party C (Service Observer)
External party releases a call		
	EventPartyDeleted ConnID 1 ThisDN B ThisDNRole Destination OtherDN A OtherDNRole DeletedParty ThirdPartyDNRole Observer ^a ThirdPartyDN C ^a CallState OK	EventPartyDeleted ConnID 1 ThisDN C ThisDNRole Observer OtherDN A OtherDNRole DeletedParty CallState OK

Monitoring calls

	EventReleased ConnID 1 ThisDN B ThisDNRole Destination OtherDN C CallState OK	EventReleased ConnID 1 ThisDN C ThisDNRole Observer OtherDN B CallState OK
--	---	--

a. The attribute contains observer information.

Observer releases first

Party A (External)	Party B	Party C (Service Observer)
		Observer releases a call
	EventPartyDeleted ConnID 1 ThisDN B ThisDNRole Destination OtherDN C OtherDNRole Observer ThirdPartyDNRole DeletedBy ThirdPartyDN C CallState OK	EventReleased ConnID 1 ThisDN C ThisDNRole Observer CallState OK
Release phase (ConnID 1)		

Abnormal call flow

Interruption point	Party A	Party B	Party C
*		EventAbandoned ConnID 1 ThisDN B OtherDN A CallState OK	

Service observing for agent-initiated call

The following table describes service observing for an agent-initiated call.

Party A	Party B	Party C
EventRinging ConnID 1 ThisDN A ThisDNRole Destination OtherDN B OtherDNRole Origination CallState OK	EventDialing ConnID 1 ThisDN B ThisDNRole Origination OtherDN A OtherDNRole Destination CallState OK	
Answer		

EventEstablished ConnID 1 ThisDN A ThisDNRole Destination OtherDN B OtherDNRole Origination CallState OK	EventEstablished ConnID 1 ThisDN B ThisDNRole Origination OtherDN A OtherDNRole Destination CallState OK	
EventPartyAdded ConnID 1 ThisDN A ThisDNRole Destination OtherDN C OtherDNRole Observer CallState Bridged	EventPartyAdded ConnID 1 ThisDN B ThisDNRole Origination OtherDN C OtherDNRole Observer CallState Bridged	EventRinging ConnID 1 ThisDN C ThisDNRole Observer OtherDN B OtherDNRole Origination CallState Bridged
		Answer
		EventEstablished ConnID 1 ThisDN C ThisDNRole Observer CallState Bridged
Conference		
Release phase (for more information, see descriptions in Service observing on agent)		

Service observing on queue

The following table describes service observing on the queue.

Party A (External)	Party B	Party C	Party D (Observer)
Inbound call			
	EventQueued ConnID 1 ThisDN B ThisDNRole Destination OtherDN A OtherDNRole Origination		EventRinging ConnID 1 ThisDN D ThisDNRole Observer OtherDN A OtherDNRole Origination CallState Bridged
			EventEstablished ConnID 1 ThisDN D ThisDNRole Observer OtherDN A OtherDNRole Origination CallState Bridged
	EventDiverted ConnID 1 ThisDN B	EventRinging ConnID 1 ThisDN C	

	ThisQueue B ThisDNRole Destination OtherDN A OtherDNRole Origination ThirdPartyDN C *OPT ThirdPartyDNRole Destination *OPT	ThisDNRole Destination OtherDN A OtherDNRole Origination CallState Bridged	
		Answer (TAnswerCall)	
		EventEstablished ConnID 1 ThisDN C ThisDNRole Destination OtherDN A OtherDNRole Origination CallState Bridged Extensions: OrigDN-1=A OrigDN-2=D	EventPartyAdded ConnID 1 ThisDN D ThisDNRole Observer OtherDN C OtherDNRole NewParty ThirdPartyDN C ThirdPartyDNRole AddedBy CallState Bridged
Conference			
Release phase (for more information, see descriptions in Service observing on agent)			

Abnormal call flow

Interruption point	Party A	Party B	Party C	Party D
*			EventAbandoned ConnID 1 ThisDN C OtherDN A CallState OK	EventReleased ConnID 1 ThisDN D ThisDNRole Observer OtherDN A CallState OK

Working with queues

Contents

- 1 Multiple-queue call treated at an IVR port with treatment at IVR queue
 - 1.1 Abnormal call flow
- 2 Multiple-queue, call treated at an IVR port with direct treatment at IVR port
 - 2.1 Abnormal call flow
- 3 Multiple-queue call with call removed from queue

Learn about call models for queues.

Related documentation:

-
-
-

RSS:

- [For private edition](#)

For simplicity, the examples on this page use abbreviated attribute values. For example, ConnID **1**, which – in actual events – displays as ConnID>@metainformation>.

The following comments and abbreviations are used in the call models:

- OPT-Optional.
- DIAL-Might be a dialed number or is not present if Voice Microservices have no information about the other party.

Multiple-queue call treated at an IVR port with treatment at IVR queue

The following table describes a multiple-queue call treated at an IVR port with treatment at the IVR queue.

A	Q1	Q2	Q3	IVR	Agent
Inbound/ internal call to Q1	Call to Q1				
EventDialing ConnID 1 ThisDN A ThisDNRole Origination OtherDN* Q1 OtherDNRole Destination					
	EventQueued ConnID 1 ThisDN Q1				

	ThisQueue Q1 OtherDN A				
		Call placed in second queue			
		EventQueued ConnID 1 ThisDN Q2 ThisQueue Q2 OtherDN A			
			Call placed in IVR queue for treatment when no agents ready		
			EventQueued ConnID 1 ThisDN Q2 ThisQueue Q2 OtherDN A		
			EventDiverted ConnID 1 ThisDN Q3 ThisQueue Q3 OtherDN A ThirdPartyDN IVR DN CallState ConverseOn	EventRinging ConnID 1 ThisDN IVR ThisQueue Q3 OtherDN A CallState ConverseOn	
				Answer	
				EventEstablished ConnID 1 ThisDN IVR ThisQueue Q3 OtherDN A	
					Agent Ready
	EventDiverted ConnID 1 ThisDN RQ2 ThisQueue RQ2 OtherDN A ThirdPartyDN AgentDN	EventDiverted ConnID 1 ThisDN Q2 ThisQueue Q2 OtherDN A ThirdPartyDN AgentDN		EventReleased a ConnID 1 ThisDN IVR ThisQueue Q3 OtherDN A	EventRinging ConnID 1 ThisDN AgentDN ThisQueue Q1 OtherDN A
					Answer
EventEstablished b ConnID 1					EventEstablished ConnID 1 ThisDN AgentDN

ThisDN A OtherDN AgentDN CallState OK					ThisQueue Q1 OtherDN A CallState OK
--	--	--	--	--	--

- a. EventReleased can occur before an agent becomes available because the IVR finishes call treatment.
- b. In some deployments, EventEstablished for party A can occur at the same time as the IVR EventEstablished, especially if a call comes through the PSTN.

Abnormal call flow

Interruption Point	A	Q1	Q2	Q3	IVR	Agent
*	EventReleased OtherDN Q1	EventAbandoned ConnID 1 ThisDN Q1 OtherDN A	EventAbandoned ConnID 1 ThisDN Q2 OtherDN A	EventAbandoned ConnID 1 ThisDN Q3 OtherDN A	EventReleased ConnID 1 ThisDN IVR OtherDN A	

Multiple-queue, call treated at an IVR port with direct treatment at IVR port

The following table describes a multiple-queue call treated at an IVR port with direct treatment at the IVR queue.

External party	Q1	Q2	IVR	Agent
Inbound/internal call to Q1	Call to Q1			
EventDialing ConnID 1 ThisDN A ThisDNRole Origination OtherDN* Q1 OtherDNRole Destination				
	EventQueued ConnID 1 ThisDN Q1 ThisQueue Q1 OtherDN A			
		Call placed in second queue		
		EventQueued		

		ConnID 1 ThisDN Q2 ThisQueue Q2 OtherDN A		
			Call placed directly to IVR port	
			EventRinging ConnID 1 ThisDN IVR OtherDN A CallState ConverseOn	
			Answer	
			EventEstablished ConnID 1 ThisDN IVR OtherDN A	
				Agent ready
	EventDiverted ConnID 1 ThisDN RQ2 ThisQueue RQ2 OtherDN A ThirdPartyDN AgentDN	EventDiverted ConnID 1 ThisDN Q2 ThisQueue Q2 OtherDN A ThirdPartyDN AgentDN	EventReleased ^a ConnID 1 ThisDN IVR OtherDN A	EventRinging ConnID 1 ThisDN AgentDN ThisQueue Q1 OtherDN A
				Answer
EventEstablished ^b ConnID 1 ThisDN A OtherDN AgentDN CallState OK				EventEstablished ConnID 1 ThisDN AgentDN ThisQueue Q1 OtherDN A CallState OK

a. EventReleased can occur before an agent becomes available because the IVR finishes call treatment.

b. In some deployments, EventEstablished for party A can occur at the same time as the IVR EventEstablished, especially if a call comes through the PSTN.

Abnormal call flow

Interruption Point	External party	Q1	Q2	IVR	Agent
*	EventReleased OtherDN Q1	EventAbandoned ConnID 1 ThisDN Q1 OtherDN A	EventAbandoned ConnID 1 ThisDN Q2 OtherDN A	EventReleased ConnID 1 ThisDN IVR OtherDN A	

Multiple-queue call with call removed from queue

The following table describes a multiple-queue call with the call removed from the queue.

A	Q1	Q2	IVR	Agent
Inbound call to Q1	Call to Q1			
	EventQueued ConnID 1 ThisDN Q1 ThisQueue Q1 OtherDN A			
		Call placed in second queue		
		EventQueued ConnID 1 ThisDN Q2 ThisQueue Q2 OtherDN A		
			Call placed in third queue for treatment when no agents ready	
			EventQueued ConnID 1 ThisDN Q3 ThisQueue Q3 OtherDN A	
			Call cleared from third queue	
			EventDiverted ConnID 1 ThisDN Q3 ThisQueue Q3 OtherDN A CallState Cleared	
				Agent ready
	EventDiverted ConnID 1 ThisDN Q1 ThisQueue Q1 OtherDN A ThirdPartyDN AgentDN	EventDiverted ConnID 1 ThisDN Q2 ThisQueue Q2 OtherDN A ThirdPartyDN AgentDN		EventRinging ConnID 1 ThisDN AgentDN ThisQueue Q1 OtherDN A CallState OK
				Answer
				EventEstablished

				ConnID 1 ThisDN AgentDN ThisQueue Q1 OtherDN A CallState OK
--	--	--	--	--

Handling user data

Contents

- [1 Attaching or updating user data to internal call](#)
- [2 Attaching or updating user data to call by third party](#)

Learn how user data is handled in Voice Microservices.

Related documentation:

-
-
-

RSS:

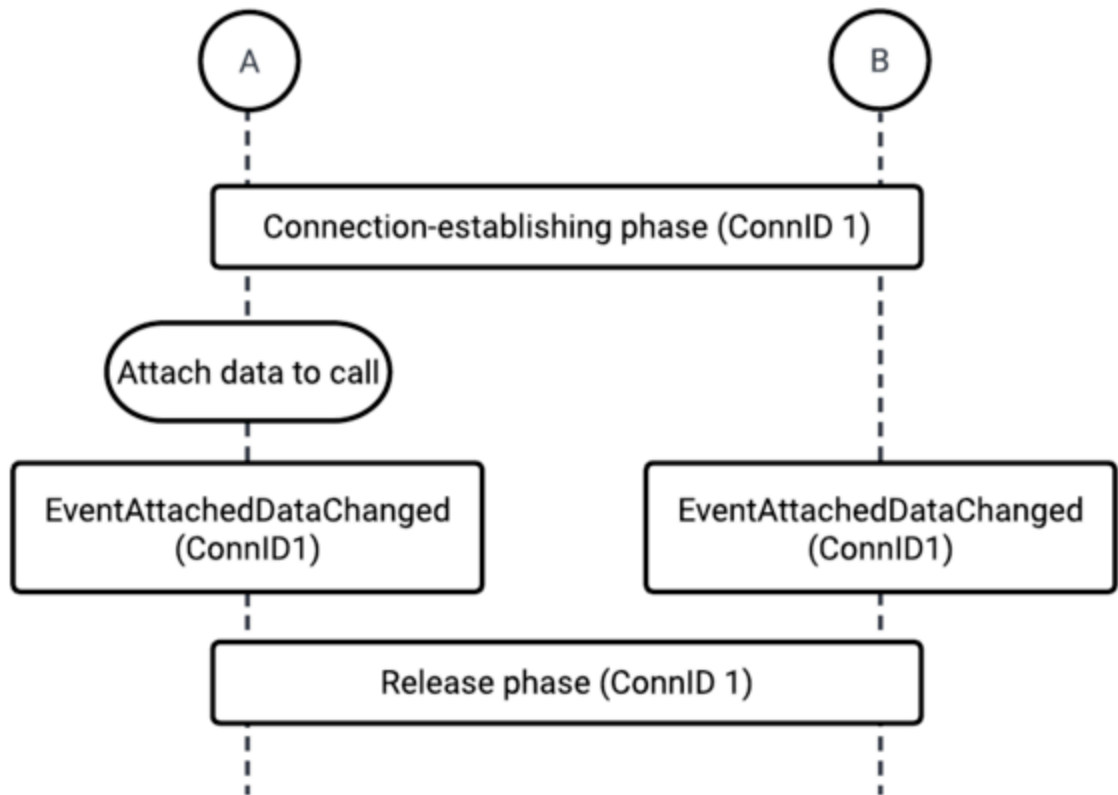
- [For private edition](#)

Voice Microservices support the scenarios described on this page. User data events are not produced to Kafka. User data attached to a call is available in call-related events.

Attaching or updating user data to internal call

The following diagram and table describe attaching/updating user data to an internal call.

For simplicity, the following example uses ConnID 1, however – in actual events – the ConnID displays as ConnID>@metainformation>.

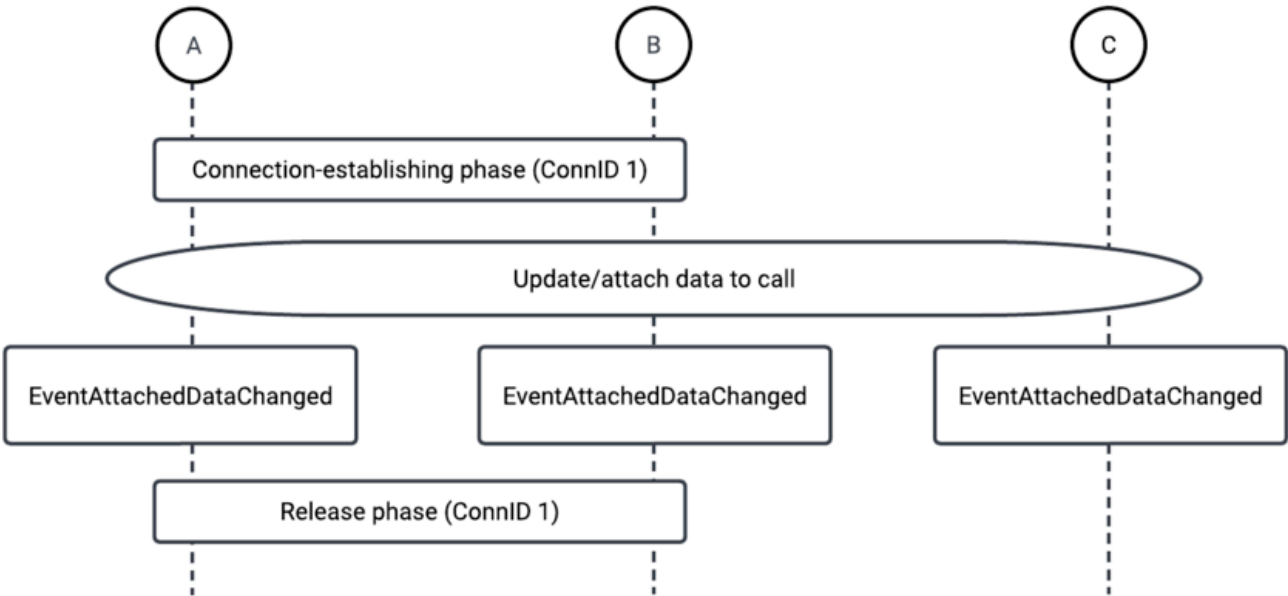


Party A	Party B
Call-establishing phase (ConnID 1)	
Attach user data to a call (TUpdateUserData)	
EventAttachedDataChanged: ConnID 1 ThisDN A ThirdPartyDN A	EventAttachedDataChanged: ConnID 1 ThisDN B ThirdPartyDN A
Release phase (ConnID 1)	

Attaching or updating user data to call by third party

The following diagram and table describe attaching/updating user data to a call by a third party.

For simplicity, the following example uses ConnID 1, however – in actual events – the ConnID displays as ConnID>@metainformation>.



Party A	Party B	Party C
Call-establishing phase (ConnID 1)		
		Attach user data to a call (TUpdateUserData)
EventAttachedDataChanged: ConnID 1 ThisDN A ThirdPartyDN C	EventAttachedDataChanged: ConnID 1 ThisDN B ThirdPartyDN C	EventAttachedDataChanged: ConnID 1 ThirdPartyDN C
Release phase (ConnID 1)		