



PureConnect®

2020 R2

Generated:

05-May-2020

Content last updated:

05-May-2019

See [Change Log](#) for summary of changes.



Interaction Tracker Interaction Segments

Technical Reference

Abstract

This content explains the improvements in interaction data collection introduced in CIC 4.0 and subsequent SUs. It provides details about the Interaction Segment types captured as part of the Interaction Segment Detail log and the reasons for the data capture.

For the latest version of this document, see the PureConnect Documentation Library at: <http://help.genesys.com/cic>.

For copyright and trademark information, see https://help.genesys.com/cic/desktop/copyright_and_trademark_information.htm.

Table of Contents

Table of Contents	2
Overview	3
Segment detail data documentation	3
CIC client	3
Schema and table detail	4
How CIC generates segments	5
Segment Details	7
End Codes	7
Segment Attributes	9
Details Attribute	9
Segment detail data in different interaction scenarios	12
Two-party calls	12
Simple ACD inbound call to workgroup queue	12
Simple ACD inbound call to IVR system	13
Internal ACD call	13
Park scenario	15
Blind transfer scenarios	15
Blind transfer an inbound call to another agent	16
Blind transfer an inbound call to an external number	16
Blind transfer a call to an agent who transfers it to another agent	18
Blind transfer an intercom call to a remote number	19
Consult transfer scenarios	21
Consult transfer an ACD Interaction to another agent	21
Consult transfer an ACD call to a remote party	24
Consult transfer an internal call to another agent	26
Consult transfer an internal call to a remote party	30
Three-party conference	33
Inbound ACD call	33
Intercom initiator	34
Intercom respondent	35
Outbound Interaction	36
Dial on behalf of workgroup outbound interaction	37
Simple outbound interaction	38
Interaction Detail API	39
Table 0 - Events	39
Table 1 - Attributes	39
Table 2 - Attribute detail data	40
Table 3 - Errors	40
Sample code	40
Miscellaneous	43
Snoozed callback and interaction details	43
Switchover	43
Multiple ACW tasks	43
Change Log	44

Overview

Before CIC 4.0, Interaction Tracker did not capture Interaction Segment Detail. It captured basic interaction information in the Call Log. The Call Log contained only a basic interaction history, due to the nature of the data collected and Call Log size limitations. getting a full interaction history was challenging.

In CIC 4.0, we added a new feature called Interaction Segment Detail data logging. This new feature captures a high-level interaction summary and details about interaction events in the new Interaction Segment Detail log. CIC captures the data in an encoded form and provides a decoding interface, the Interaction Detail View, for the customer. Using this data, Call Center managers and supervisors can view a full interaction history or create custom reports. Each segment detail entry includes StartDateTimeUTC with millisecond granularity.

Segment detail data documentation

These resources provide more information about Interaction Segment Detail.

- Information about the Segment Log URI Design is available in this KB article:
https://genesyspartner.force.com/customercare/pkb_Home?id=kA50B0000008R1jSAE&l=en_US&fs=Search&pn=1 .
- A description of Event IDs within the Segment Log is available in this KB article:
https://genesyspartner.force.com/customercare/pkb_Home?id=kA50B0000008R1oSAE&l=en_US&fs=Search&pn=1

CIC client

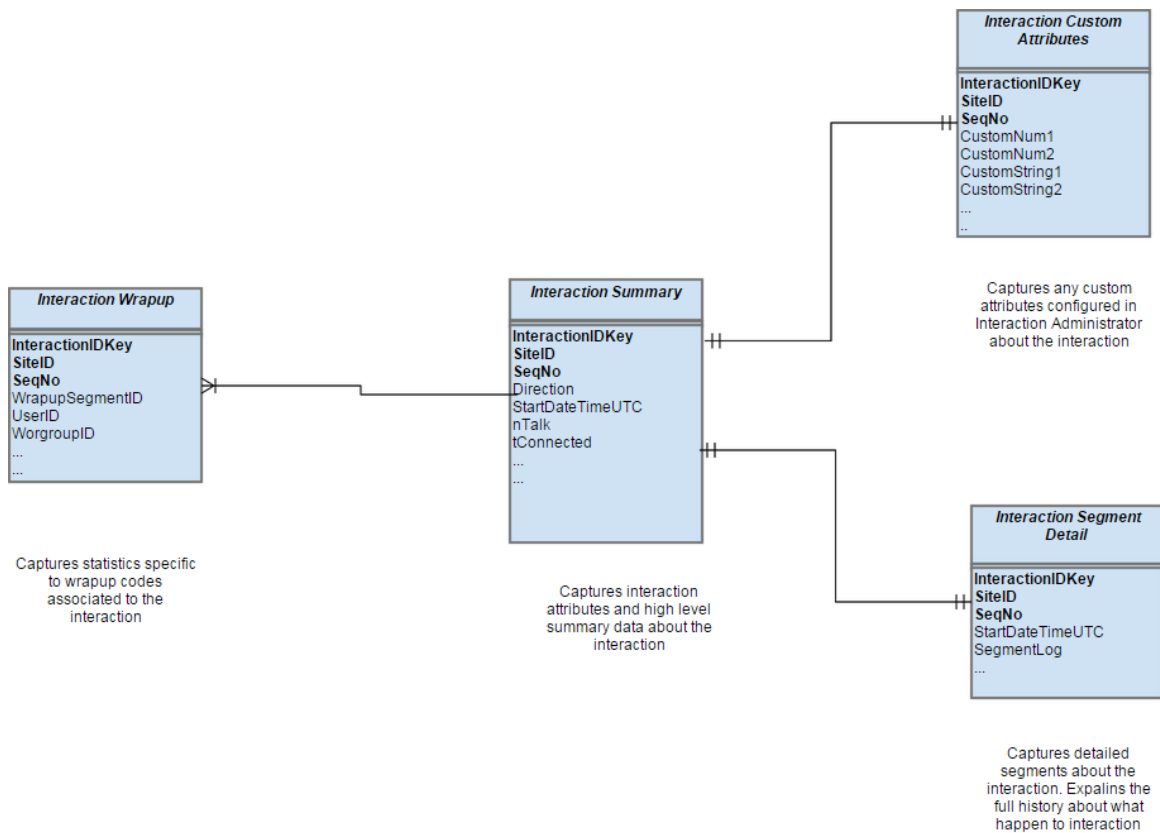
Customer Interaction Center (CIC) supports two interaction management client applications. This documentation uses the term CIC client to refer to either Interaction Connect or Interaction Desktop.

Schema and table detail

- Here is schema used before CIC 4.0.

<i>Calldetail</i>
callIdkey
SiteID
InitiatedDateTimeUTC
TerminatedDateTime
...
...
CustomNum1
CustomNum2
CustomNum3
CustomString1
....

- In 4.0 GA and later, CIC captures the interaction detail data in four tables
- These tables use an identical composite key: InteractionIDKey, SiteID, and SeqNo. You use this primary key to link the data in these tables.



Note: For more information about the attributes for these tables, see the [PureConnect Data Dictionary Technical Reference](#) in the [PureConnect Documentation Library](#).

Table	Description
Interaction Summary	This table replaces our previous Calldetail table. It includes all the interaction attributes previously captured in the Calldetail table. In addition, it includes high-level summary data about the interaction. For example, you can determine how many times participants put the interaction on hold (nHeld). It also includes the total hold time (tHeld) for the interaction during its entire life cycle.
Interaction Segment Detail	This table captures interaction history in predefined segments.
Interaction Wrap-up	If the interaction has an associated wrap-up code, this table captures all the related wrap-up code statistics.
Interaction Custom Attributes	This table replaces the custom attributes in the previous Calldetail table. It includes 10 custom columns: five number columns and five string columns. The customer can define their custom attributes just as they did before CIC 4.0. This table captures the data.

How CIC generates segments

This section explains which event generates each segment type. It describes any special attributes for each segment type. For more information about segment attributes, see the [Segment Attributes](#) section.

Note: The **Segments** node in the **Interaction Details** dialog box displays these segments in top-down chronological order.

Segment Type	Triggering Event	Description	Special attributes
Alert	The system alerts a user or station.	How long the interaction is in an alert state with its associated parties.	During an Alert event for an intercom interaction request, the interaction includes both parties of the interaction.
System	The interaction enters the system state.	How long the interaction is in the system state before it transitions to another state	A system segment can also indicate the survey operation. If the pre-survey system segment ends as survey, then the system segment indicates how long the caller interacted with our Survey questionnaire.
Queue	CIC offers an interaction to a workgroup; the system starts capturing the time. When the interaction state changes from offered to workgroup to something else (such as disconnect or offer to the user), it stops computing the duration.	How long the interaction waits in a workgroup queue before it transitions to the next state (such as alert, connect, or disconnect).	See the Segment Attributes section.
Hold	An agent puts the interaction on hold, using either a CIC client or a station-only phone.	How long the interaction is in the Hold state before it transitions to the next state	See the Segment Attributes section.
Park	An agent parks the interaction on an orbit queue, workgroup queue, or user queue, using a CIC client.	How long the interaction is in the Park state before it transitions to the next state.	See the Segment Attributes section.

Messaging	An agent does not pick up the interaction and it reaches voice mail. Or an agent selects the interaction and sends it to voice mail.	How long the interaction is in the Messaging/Voicemail state before it transitions to the next state. This segment can occur before or after an interaction connects to a CIC user or CIC station.	Not applicable
Connect	The interaction connects two parties (such as when an agent picks up an internal or external interaction.) The participants can be CIC users, CIC stations, or remote persons. CIC generates a connect segment when two parties connect with each other. For example, CIC does not generate a connect segment during a conference when all parties (3 or more) parties are in a connected state. Also, if only one side of the interaction connects, CIC does not generate a connect segment during an intercom scenario.	How long the interaction is in connected state before it transitions to the next state	See the Segment Attributes section.
Conference	This segment does not depend on interaction state. Instead CIC generates this segment when it creates the conference to connect three or more participants,	Conference participation. How long CIC connects three or more parties in a conference interaction. This segment type applies only to telephone calls.	This segment includes a special detail attribute called ConfParties. This attribute captures information in XML format about all the parties involved in the conference.
Consult	An internal user uses the Consult Call dialog in a CIC client to consult another agent or remote party on behalf of an ACD interaction. If the agent does not use the Consult Call dialog, CIC does not generate a Consult segment.	How long an internal user (agent) consults another party, One participant is an Agent, but the other one can be an Agent or a Remote Party (non-CIC user). During this consult segment, the interaction is in a connected state.	Not applicable
IVR	A call enters CIC using the out-of-the-box IVR handlers; Interaction Attendant sets the <code>EIC_IVRAppName</code> attribute. This event generates the IVR segment. When the call enters a workgroup queue. it closes the segment If the customers did not use our out-of-the-box IVR handlers, CIC might not generate an IVR segment. Customers are expected to set this attribute when it enters their IVR if they want to capture this segment. Otherwise it is logged as system segment.	How long the call is in the IVR application before it transitions to next state (such as disconnect or transfer to another user or workgroup queue.)	See the Segment Attributes section.
After Call Work	An agent's status changes to Follow-up or After Call Work after handling an ACD workgroup interaction.	How long an agent's status remains Follow-up or After Call Work after handling an ACD interaction.	Not applicable.

Dialing	An agent starts an outbound interaction or an intercom interaction. This segment captures both the Dialing and Proceeding states. CIC treats an intercom operation as two separate interactions. The agent who starts the intercom interaction has the Dialing segment. The agent who picks up the intercom interaction has the Alert segment	How long the interaction is in a Dialing state before it transitions to a Connect or Disconnect state.	Not applicable
External Transfer	An agent transfers any interaction outside the CIC system. This causes two external parties to use the CIC system to communicate. Multiple system operations such as a Blind Transfer to a Remote Number or a Consult Transfer to a Remote Number can generate this segment. When IVR uses the Remote Transfer tool step, CIC generates the external transfer segment. CIC also generates this segment when an agent configures a remote number for an Available, Forward or Available, Follow-Me status.	The duration of an explicit or implicit external transfer operation.	This segment includes some additional attributes about the transfer. See the Segment Attributes section.
Suspend	Not applicable	For future reference.	Not applicable
Unknown	CIC failed to identify this segment type.		

Segment Details

This section explains the Interaction Segment Detail log. Each segment identifies interaction events. The log captures details such as interaction type, duration, participants, and more for each segment. This table describes the different segment types that occur in different interaction scenarios. It explains which events or operations generate each segment.

[End Codes](#)

[Segment Attributes](#)

[Details Attribute](#)

End Codes

An end code describes how a segment ended.

End Code	Description
LocalDisconnect	If an agent or local party disconnects one side of an interaction, the segment associated with that agent or local party includes the LocalDisconnect end code. For example, if an agent disconnects an inbound ACD interaction after talking to a remote party, the disposition of the final Connect segment is Local Disconnect.
RemoteDisconnect	If an agent or local party connects with an external party and then the external party disconnects the interaction, the last segment's disposition is Remote Disconnect. This end code occurs because the last segment is associated with the local party and the local party did not disconnect the interaction. Otherwise, the disposition is LocalDisconnect.
Disconnect	When the interaction is not locally or remotely disconnected, the disposition is Disconnect. This disposition can occur in some conference call disconnections or when CIC disconnects the call for some reason.
LocalTransfer	If an agent transfers an interaction within the CIC system to other internal objects such as different local queues, agents, stations, or the IVR, then the disposition of the last transferred segment is LocalTransfer.
RemoteTransfer	When an agent transfers an interaction out of CIC system to a remote number or another CIC Server within a multi-site environment, the disposition of the last transferred segment is RemoteTransfer.
Conference	When the following segment ends as Conference, the disposition of the pre-conference segment is Conference.
System	When an interaction enters the System state, the disposition of the previous segment is System.
Interact	When an interaction ends in a Connected state with two parties connected in the interaction, then the disposition of the previous segment is Interact. CIC generates an Interact segment when an interaction connects with two parties.
Messaging	When an interaction reaches the voice mail system or the interaction state changes to the Messaging state, the disposition of the previous segment is Messaging.
Alert	When an interaction alerts a CIC user or station, the disposition of the previous segment is Alert.
Hold	When a party uses a CIC client or a station to put an interaction on hold, the previous segment ends with the Hold end code.
Grabbed	Supervisors and other users with the appropriate rights can <i>grab</i> interactions from other users. If another user grabs an interaction to which Agent1 is connected, then the last segment associated with Agent1 has the Grabbed end code.
IVR	When an agent transfers an interaction to the IVR, the previous segment has the IVR end code.
Queue	When an interaction is ACD routed or transferred to a workgroup queue, the previous segment has the Queue end code.
Survey	When an interaction reaches the post-survey operation, the previous segment has the Survey end code.
Snoozed	This end code applies only to the Callback media type. When the snoozing attribute (<code>EIC_snoozed</code>) is set in the System segment of a Callback media type, the System segment ends as Snoozed.
NotEnded	If, for some reason, the segment did not close correctly, the segment end code is NotEnded.
Park	If an agent parks an interaction on another queue, the previous segment has the Park end code.
Suspend	For future use.

Segment Attributes

Each event in an interaction is contained in a segment. Each segment contains the associated event attributes. This table describes all the possible attributes for the different segment types.

Attribute Name	Description
Interaction Idkey	The interaction's unique ID
Segment Sequence	This attribute can be 0. Used only when the time stamp of two successive Segments is the same.
Duration	Duration in milliseconds
Local Party	The internal user or station associated with the interaction. Certain segment types can have more than one associated local party. For example, an <i>Alert</i> segment can have two parties for an intercom call. And also a Conference segment can have three or more associated parties.
Queue	The queue targeted for the interaction. If there is no queue, this attribute is empty.
End Code	Code for how the event ended. For more information, see the End Codes section.
Details	See the Details Attribute section.
Client UTC Time	This attribute is the start time of the segment.
WrapupCode	If a Wrap-up code is associated with this segment, this attribute contains the code as defined in Interaction Administrator.
WrapupUpSegment Number	Sequence number associated with the Wrap-up code. You can use this attribute in a SQL Join query against the Wrap-up Segment table.
RemoteAddress	Remote phone number.
RemoteName	Remote party name.

Details Attribute

The Details attribute contains more attributes. This section explains how these attributes apply to different segment types for different media types. These additional attributes first appeared in CIC 4.0 SUs (Service Updates) as noted in the following table.

Attribute name	Description	Segment Type	Media Type	4.0 SU
OtherInteractionId	When CIC transfers a call to a remote number, it generates a new InteractionID that represents the outbound external call. The inbound call contains an External Transfer segment for the remaining call duration. This segment's OtherInteractionId attribute contains the InteractionIDkey of the new outbound external call.	External Transfer	Call	SU2
RemoteNumber TransferredTo	When CIC transfers a call to a remote number, the call's External Transfer segment includes the RemoteNumberTransferredTo attribute. This attribute contains the number to which CIC transferred the inbound call.	External Transfer	Call	SU2

Available-Follow-me/Forward	When CIC redirects an interaction because an agent's status is Available, Forward or Available, Follow-Me, it sets this attribute to Yes. CIC sets this attribute on both the inbound interaction and outbound made to reach the forwarded number	Connect External Transfer	Call	SU2
TransferringInteractionId	In a blind transfer scenario, TransferringInteractionId stores the InteractionID that represents the original inbound call.	Pre-Transfer segments: Alert System Queue Connect Held	Call	SU4
TransferredInteractionId	For a blind transfer scenario, TransferredInteractionId contains the InteractionID representing the original inbound call which an agent transferred to another agent or external number.	Pre-Transfer segments: Alert System Queue Connect Held	Call	SU4
TransferringUser	For a blind transfer scenario, TransferringUser contains the ICUserID for the user performing the transfer operation.	Pre-Transfer segments: Alert System Queue Connect Held	Call	SU4
TransferredUser	For a blind transfer scenario, TransferredUser contains the ICUserID of the user who started the now transferred interaction. For example, if Agent1 calls Agent2 and Agent 2 transfers the call to Agent3, Agent1 is the TransferredUser in the Pre-Transfer segment. Agent2 is the TransferringUser.	Pre-Transfer segments: Alert System Queue Connect Held	Call	SU4
IVRAppName	When a call enters the IVR, the call's IVR segment includes the IVRAppName attribute. This attribute contains the name of the IVR profile that serviced the interaction.	IVR	Call	SU4

ACDSkillSet	When an ACD call carries a specific skill set value, then segments to which this skill set value applies, store this value. Tracker Server also captures the time stamp when the skill set value is set on an interaction. For a segment where a new skill set value replaces an old value, that segment contains both the old and the new value. The segment has the old value for the initial period and the new value for the rest of the segment duration.	Alert System Queue Connect Held	Call	SU5
SecuredSessionStartTime AndDuration	This attribute tracks secured sessions during a segment. If one or more secured IVR sessions occur within a segment boundary, then the sessions start time and duration are available in this attribute. For a secured session, Interaction Tracker stores startdatetime followed by a space and then duration in seconds followed by a comma. If there are multiple secured sessions during a segment, this attribute contains multiple values with the values for each session separated by a comma.	Connect System	Call	SU5
SnoozeStartTime	An agent can <i>snooze</i> an alerting callback. This action removes the callback request from the agent's queue and places it back on the workgroup queue in a "Snoozed" state. This attribute records when the agent <i>snoozed</i> the callback.	System	Callback	SU5
SnoozeDuration InSeconds	This attribute records how long the callback remained in a <i>snoozed</i> state.	System	Callback	SU5
CallbackRetries	Number of times the agent attempted to fulfill the callback request.	System	Callback	SU5

Segment detail data in different interaction scenarios

Two-party calls

Simple ACD inbound call to workgroup queue

An external party phones the Call Center's 800 number and directly reaches a workgroup queue without going through the IVR. A Call Center agent picks up the ACD inbound interaction.

Because the external party did not use our out-of-box IVR handler or Attendant node, CIC doesn't capture an IVR segment. Instead CIC captures a System segment for the call.

Interaction Details for 1001838811

Inbound Call 4/21/2014 1:55 PM
Interactive Intelligence - sip:3177158491@inin.com:5060

START 0s 10s 20s 24.8s END

Interaction Summary

Interaction ID	1001838811
Type	Call
Duration	00:00:24
Time	4/21/2014 1:55 PM
Disposition	Remote Disconnect during Connect

Direction	Inbound
Remote Address	sip:3177158491@inin.com:5060
DNIS	sip:3@sg-clay24:5060

Recorded	Yes
Surveyed	No

Remote Party	Interactive Intelligence
Local Parties	sg-clay24_user

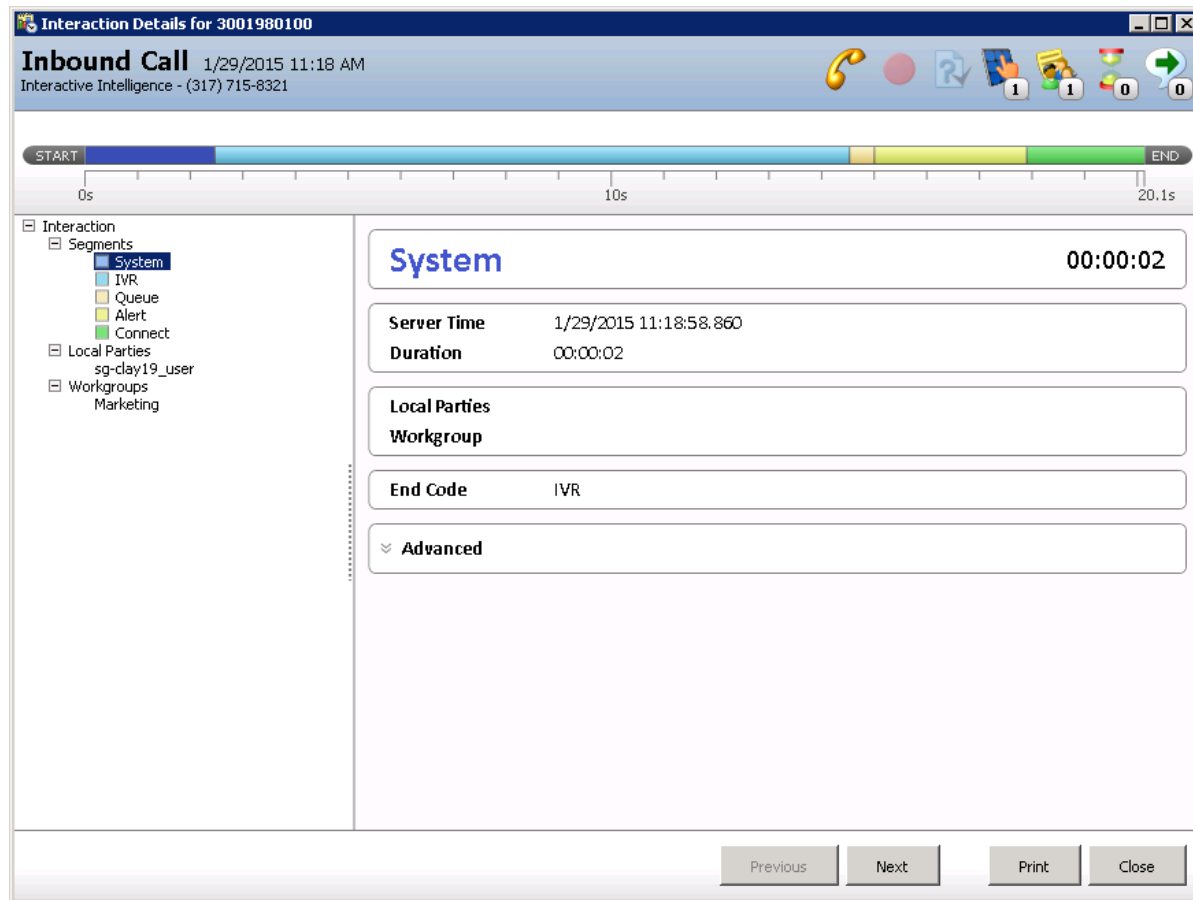
≡ Call Log

≡ Advanced

Previous Next Print Close

Simple ACD inbound call to IVR system

An external party phones the Call Center's 800 number and enters the IVR system. CIC routes the call to the appropriate workgroup. In this example, the system uses our default IVR handlers or Attendant node, so the system creates an IVR segment in addition to a System segment.

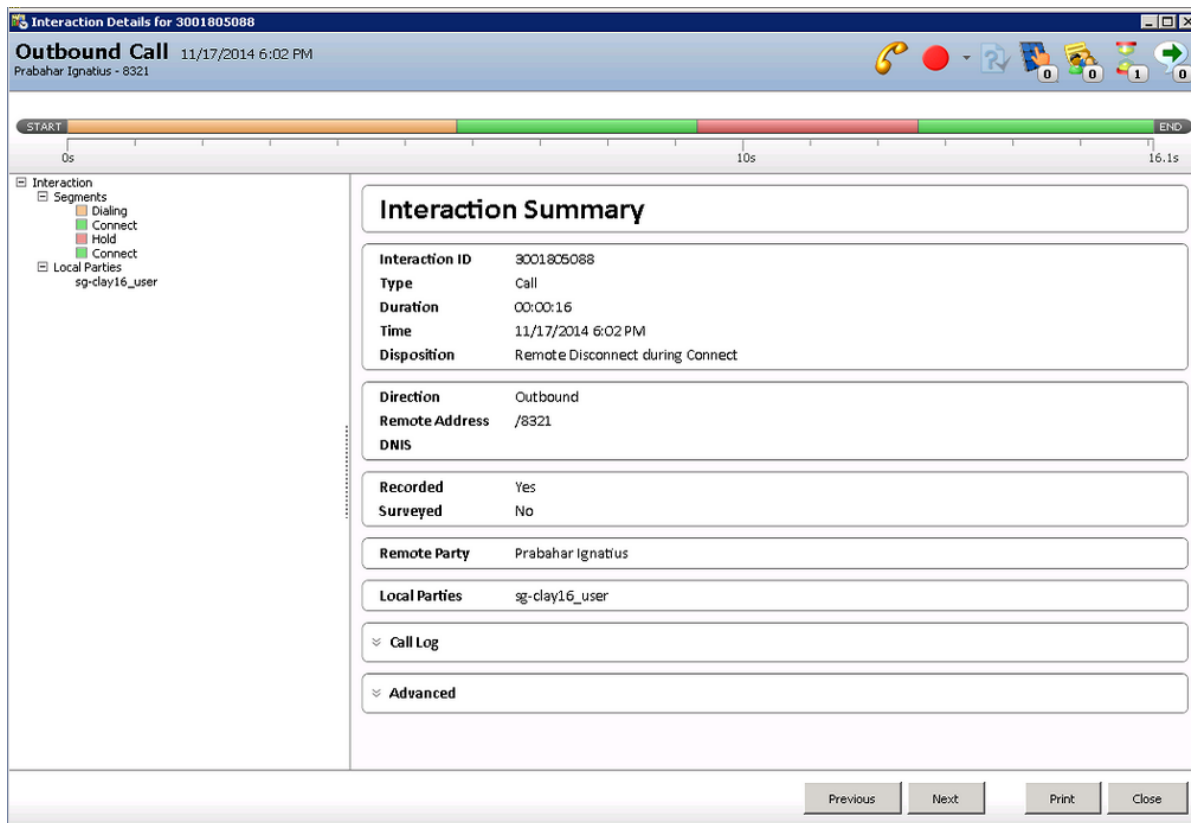


Internal ACD call

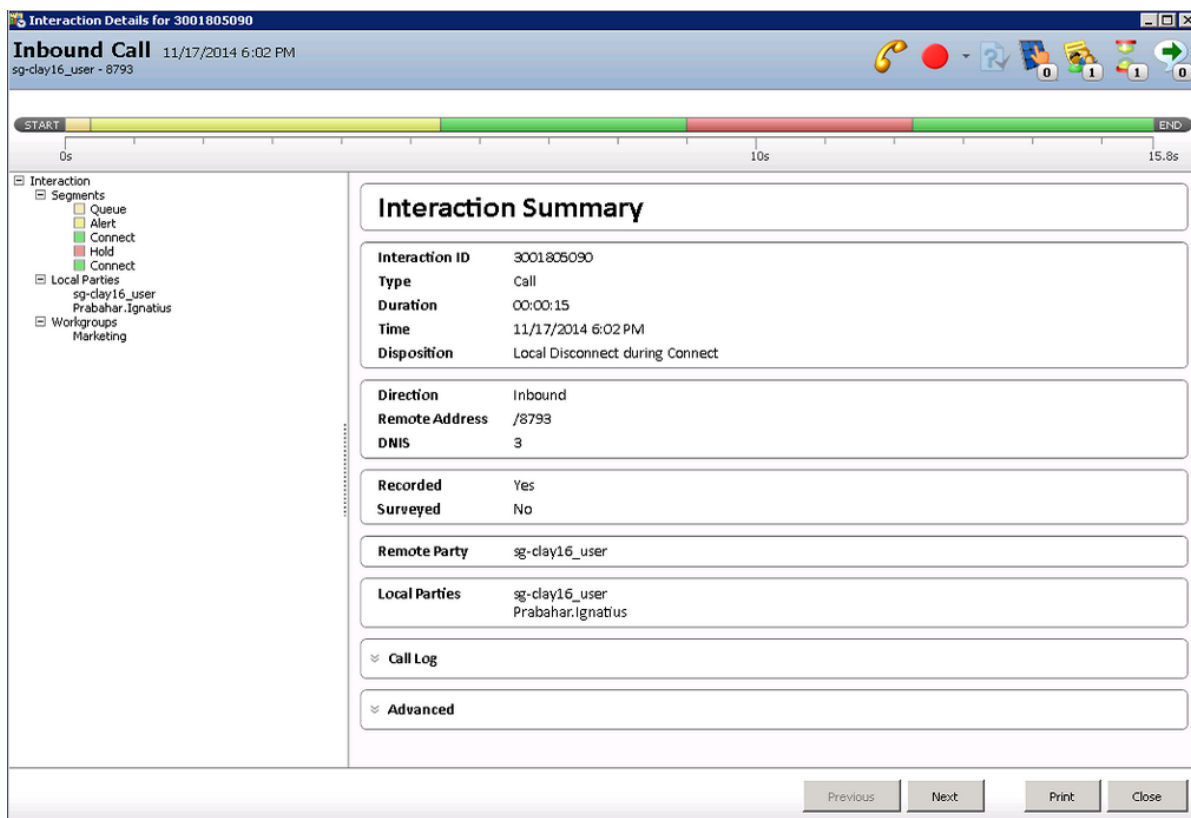
Agent1 calls one of the workgroup queues. Agent2 from the workgroup queue answers Agent1. In this scenario, there are two InteractionIDs. One for the initiator agent side (Agent1) and one for the respondent agent side (Agent2).

- The direction of **initiator** side of the intercom interaction is **outbound**.
- The direction of the **respondent** side of the intercom interaction is **inbound**.

Intercom initiator



Intercom respondent



Park scenario

An external party calls Agent1. Agent1 picks up the interaction, puts it on hold and then parks the call on Agent2's queue. The call remains on Park for a while and then Agent2 picks up. This creates a Park segment.

Interaction Tracker began to track Park segments in CIC 2016 R3.

- Prior to this change, Tracker legacy tables (`IntxSegment` and `Intx_participant`) showed that the Hold segment ended in Park (`HowEnded=14`). The next segment was the Connect segment between the caller and Agent 2. After the change, the Hold segment still ends in Park, but the next segment is the Park segment (`SegmentType=4096`). This is followed by the Connect segment between the call and Agent 2.
- After the change, the `InteractionSegmentDetail` table can now have a Park segment in the **SegmentLog** column. When a Park segment is present in the **SegmentLog** column, the Park segment appears in the **Interaction Detail Viewer** as explained in the following example.

In this example, a call alerts, is connected, put on hold, and then parked on the **Anthony.Smith** queue.

The screenshot displays the 'Interaction Details for 2001680013' window. The top bar indicates an 'Inbound Call' on 9/17/2015 at 10:37 AM, with the source 'Interactive Intelligence - sip:3177158491@i3domain.inin.com:5060'. A timeline at the top shows the call duration from 0s to 1:18, with segments for System, Alert, Connect, Hold, Park, and Connect. The left sidebar lists the interaction segments and local parties: System, Alert, Connect, Hold, Park, Connect, and Local Parties (Ajay.Bhargava, Anthony.Smith). The main content area shows the 'Park' segment with a duration of 00:00:14. Below this, the 'Server Time' is 9/17/2015 10:38:24.907, and the 'Duration' is 00:00:14. The 'Local Parties' section lists 'Anthony.Smith' and the 'Workgroup' is 'Interact'. The 'End Code' is 'Interact'. An 'Advanced' section contains a table with the following data:

Key	Value
InteractionKey	200168001320150917
SegmentSequence	0
Duration	14275
LocalParty	user queue:Anthony.Smith
Queue	
EndCode	Interact
Details	
ClientUtc	20150917T104202.174-04
SegmentSequence	0

At the bottom of the window are buttons for 'Previous', 'Next', 'Print', and 'Close'.

Blind transfer scenarios

The following scenarios illustrate the different types of blind transfer operations and their associated segments captured in the Segment Log. The Transfer icon indicates when the interaction was transferred.

[Blind transfer an inbound call to another agent](#)

[Blind transfer an inbound call to an external number](#)

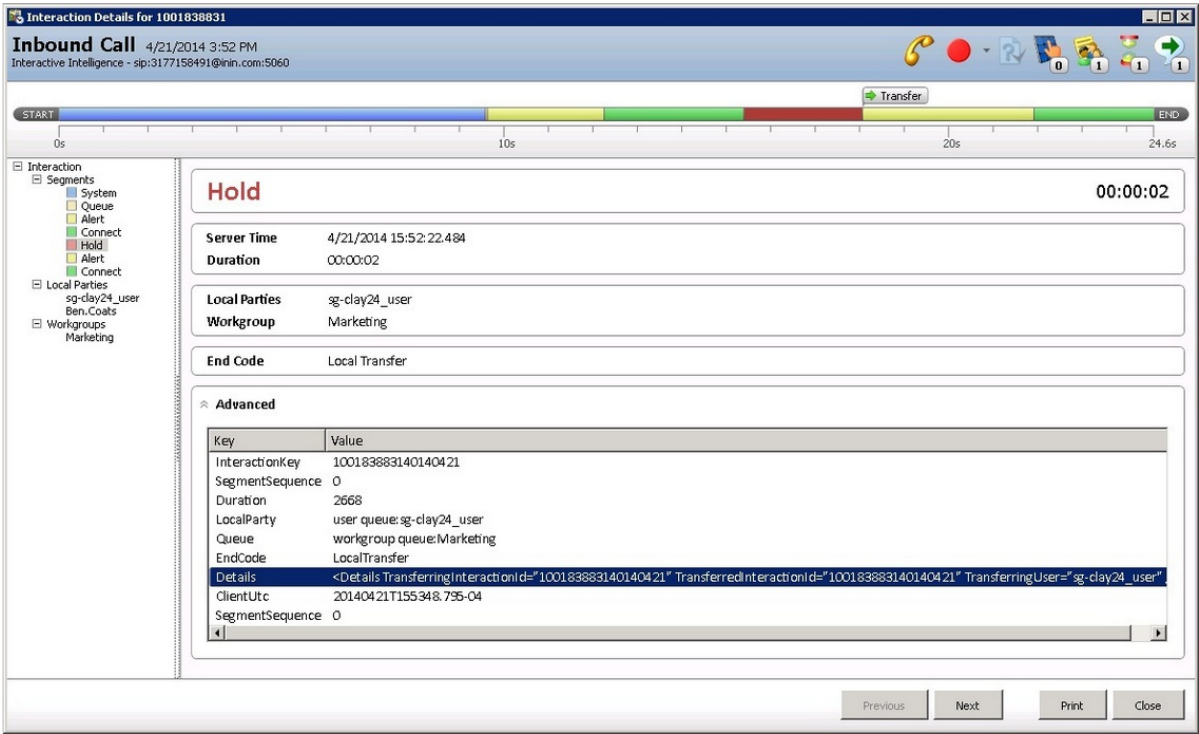
[Blind transfer a call to an agent who transfers it to another agent](#)

[Blind transfer an intercom call to a remote number](#)

Blind transfer an inbound call to another agent

Agent1 receives a call from an external party. Agent 1 puts the call on hold and then transfers the call to another agent (Agent2). There is only one InteractionID (InteractionID) in this scenario. The InteractionID captures the entire interaction history.

Note: There are some additional details captured in the transferred segment. These details include: TransferringInteractionId, TransferredInteractionId, TransferringUser, and TransferredUser. For more information, see the preceding [Details Attribute](#) section.



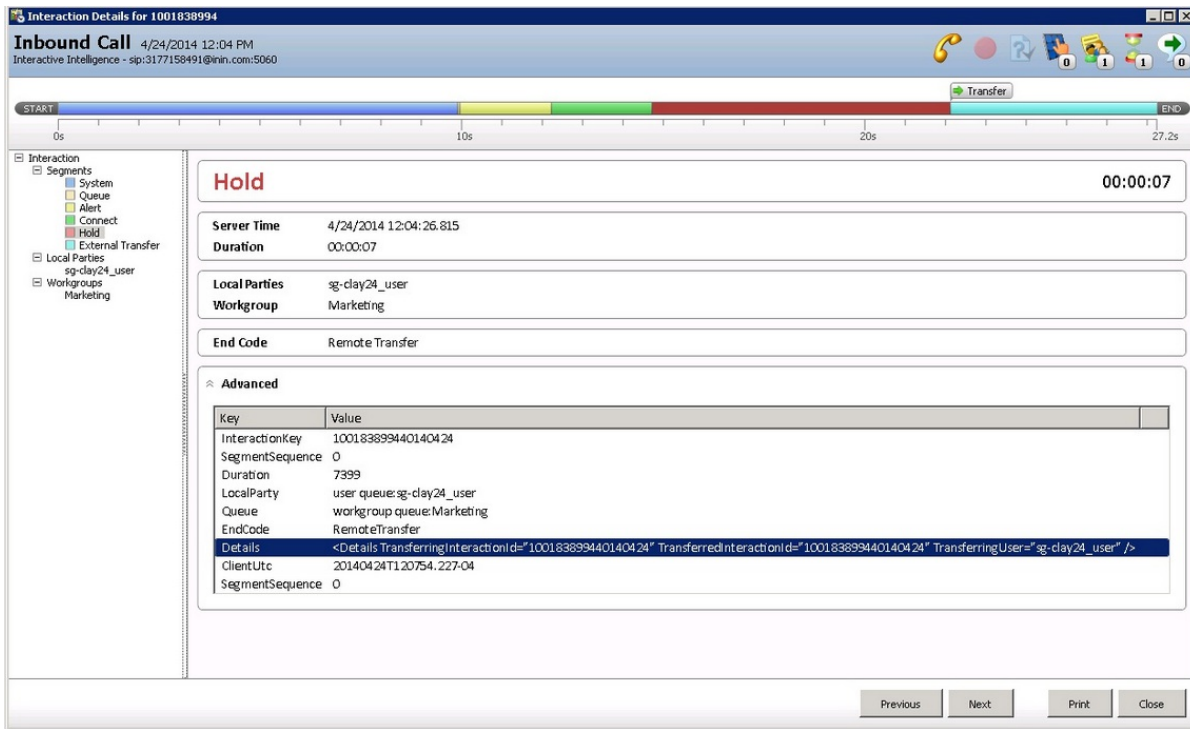
Blind transfer an inbound call to an external number

Agent1 receives a call from an external Party. Then Agent1 puts the interaction on hold and transfers the remote party to an external number. (The interaction is transferred out of the CIC system and answered by another agent.)

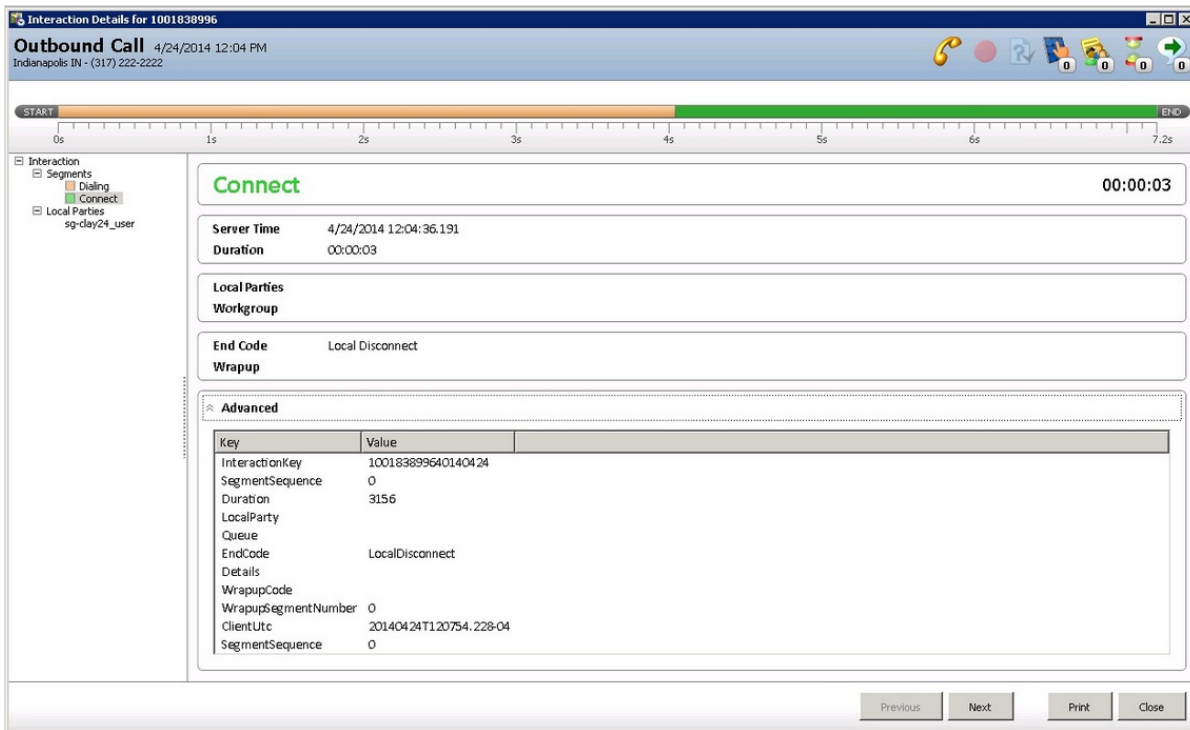
In this scenario, there are two InteractionIDs. One for the inbound call and one for the outbound call made to the external number. The inbound InteractionID captures the transfer history.

Note: The pre-transfer segment (the *Hold* segment) has some additional attributes about the transfer operation. The external transfer segment at the end indicates that the interaction is transferred out of the CIC system.

Inbound call



Outbound call



Blind transfer a call to an agent who transfers it to another agent

Agent1 receives a call from Agent2. Agent1 puts the call on hold and transfers the call to Agent3. Agent2 and Agent3 talk for few minutes and disconnect.

In this scenario, there are two InteractionIDs, one for the initiator agent side (Agent2) and one for the respondent agent side (Agent1). When Agent1 transfers the call to Agent3, the intercom initiator interaction InteractionID contains the history of entire interaction.

Note: The pre-transfer segment contains more attributes about the transfer operation. It includes the agent who transferred the interaction (`TransferringUser`) and also the agent who received the transferred interaction (`TransferredUser`).

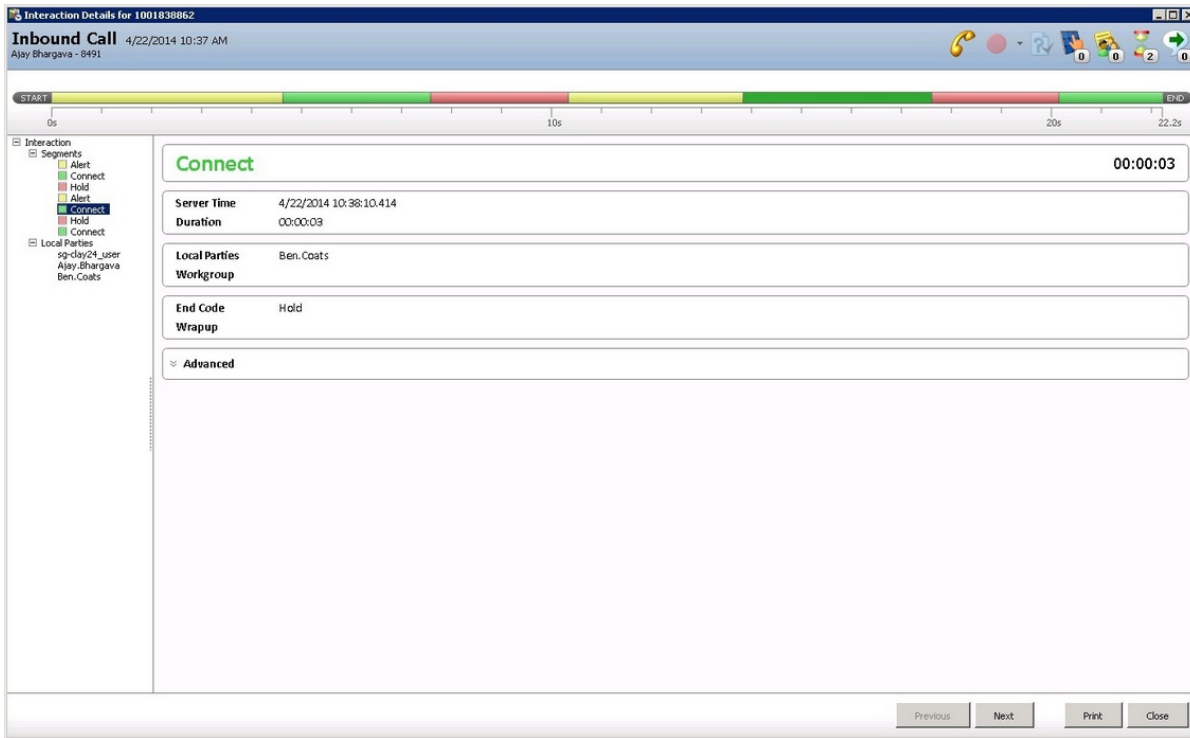
Initiator

The screenshot displays the 'Interaction Details for 1001838866' window. The title bar indicates it's an 'Outbound Call' from 'Ben Coats - 4451' on '4/22/2014 10:59 AM'. The main area shows a timeline from 0s to 24.1s, with a 'Hold' segment highlighted in red. Below the timeline, the 'Hold' status is confirmed with a duration of '00:00:07'. The 'Server Time' is '4/22/2014 10:59:11.289' and the 'Duration' is '00:00:07'. The 'Local Parties' section shows 'sg-clay24_user' as the 'Workgroup'. The 'End Code' is 'Remote Transfer'. The 'Advanced' section contains a table with the following data:

Key	Value
InteractionKey	100183886640140422
SegmentSequence	0
Duration	7765
LocalParty	user queue:sg-clay24_user
Queue	
EndCode	RemoteTransfer
Details	<Details TransferringInteractionId="100183886640140422" TransferringUser="Ben Coats" TransferredInteractionId="100183886640140422" TransferredUser="sg-clay24_user" />
ClientUtc	20140422T110035.296-04
SegmentSequence	0

At the bottom of the window, there are buttons for 'Previous', 'Next', 'Print', and 'Close'.

Respondent



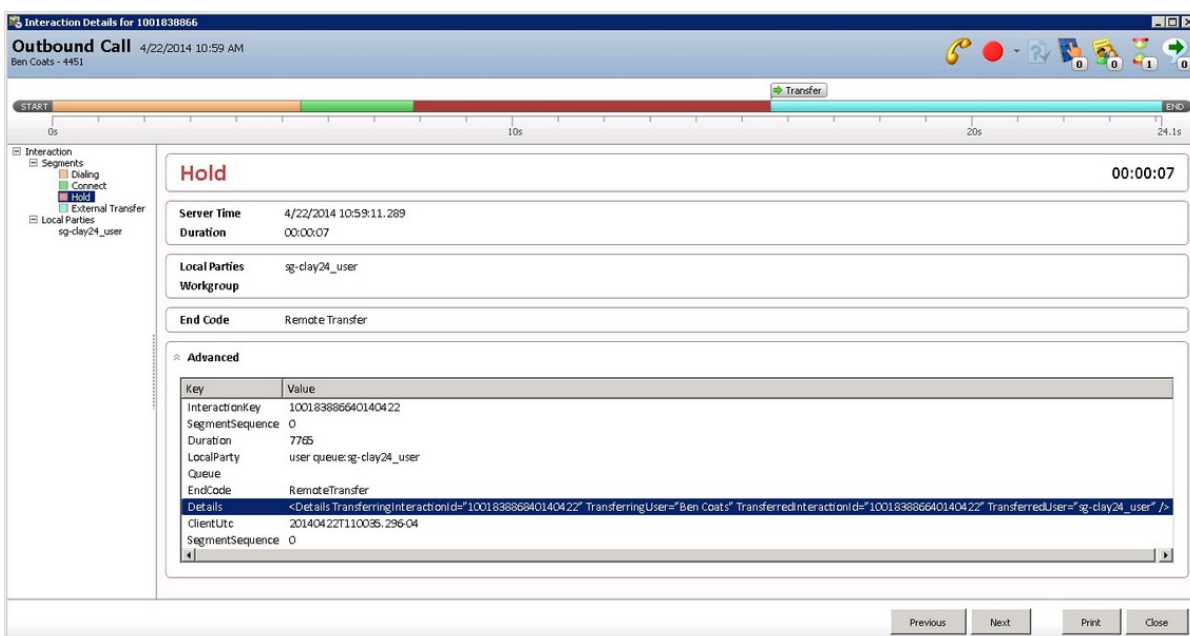
Blind transfer an intercom call to a remote number

Agent1 receives a call from Agent2. Then Agent1 puts the call on hold and transfers the call to a remote number.

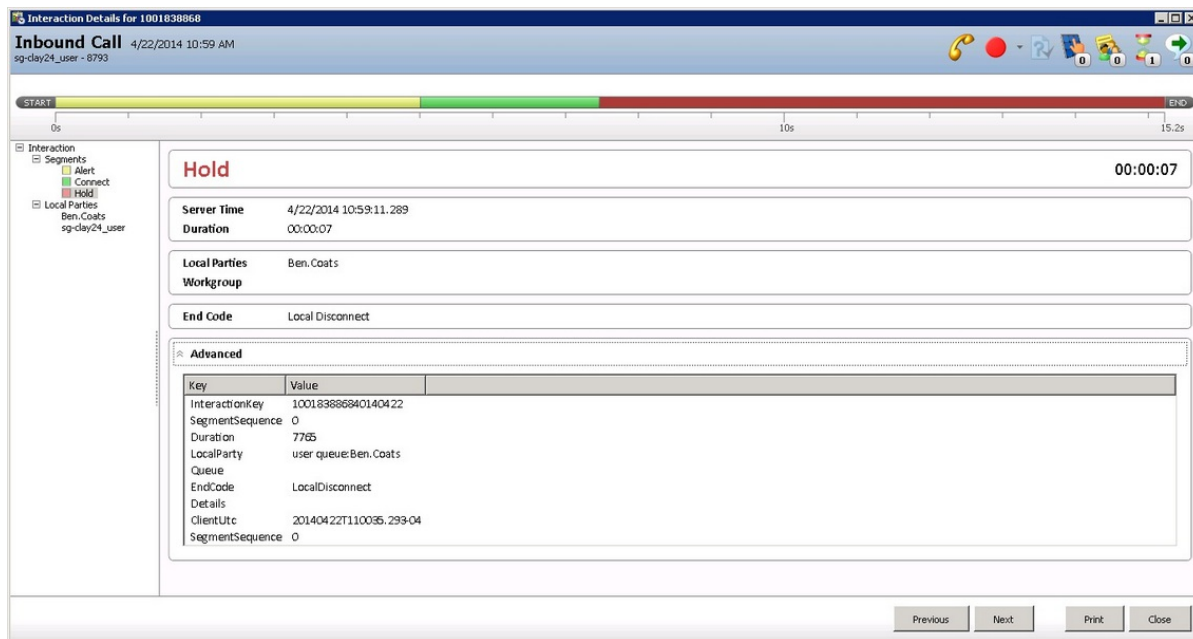
In this scenario, there are three InteractionIDs: initiator side (Agent2), respondent side (Agent1), and the outbound external call. The initiating InteractionID captures the entire history of the transfer. The initiating InteractionID is the one remaining in the system after the transfer is successful. The other two InteractionIDs are disconnected after the transfer operation completes.

Note: The pre-transfer segment contains the same transfer options detail as the [Blind transfer a call to an agent who transfers it to another agent](#) scenario.

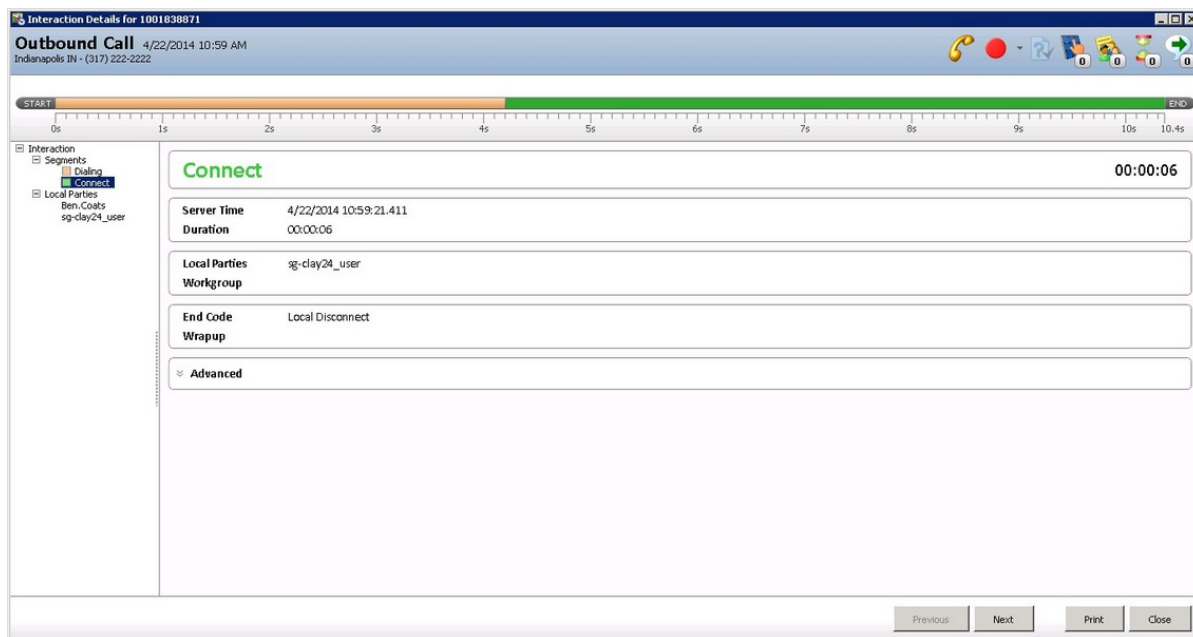
Initiator



Respondent

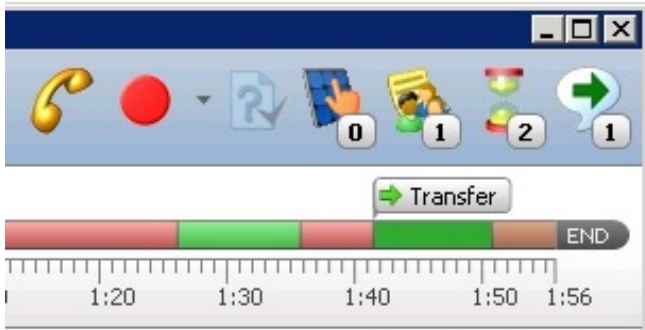


Outbound external



Consult transfer scenarios

The following sections illustrate the different types of consult calls and their associated segments captured in interaction Segment Log. The transfer icon above the timeline at the top of the **Interaction Details** dialog identifies the transferred segment.



[Consult transfer an ACD Interaction to another agent](#)

[Consult transfer an ACD call to a remote party](#)

[Consult transfer an internal call to another agent](#)

[Consult transfer an internal call to a remote party](#)

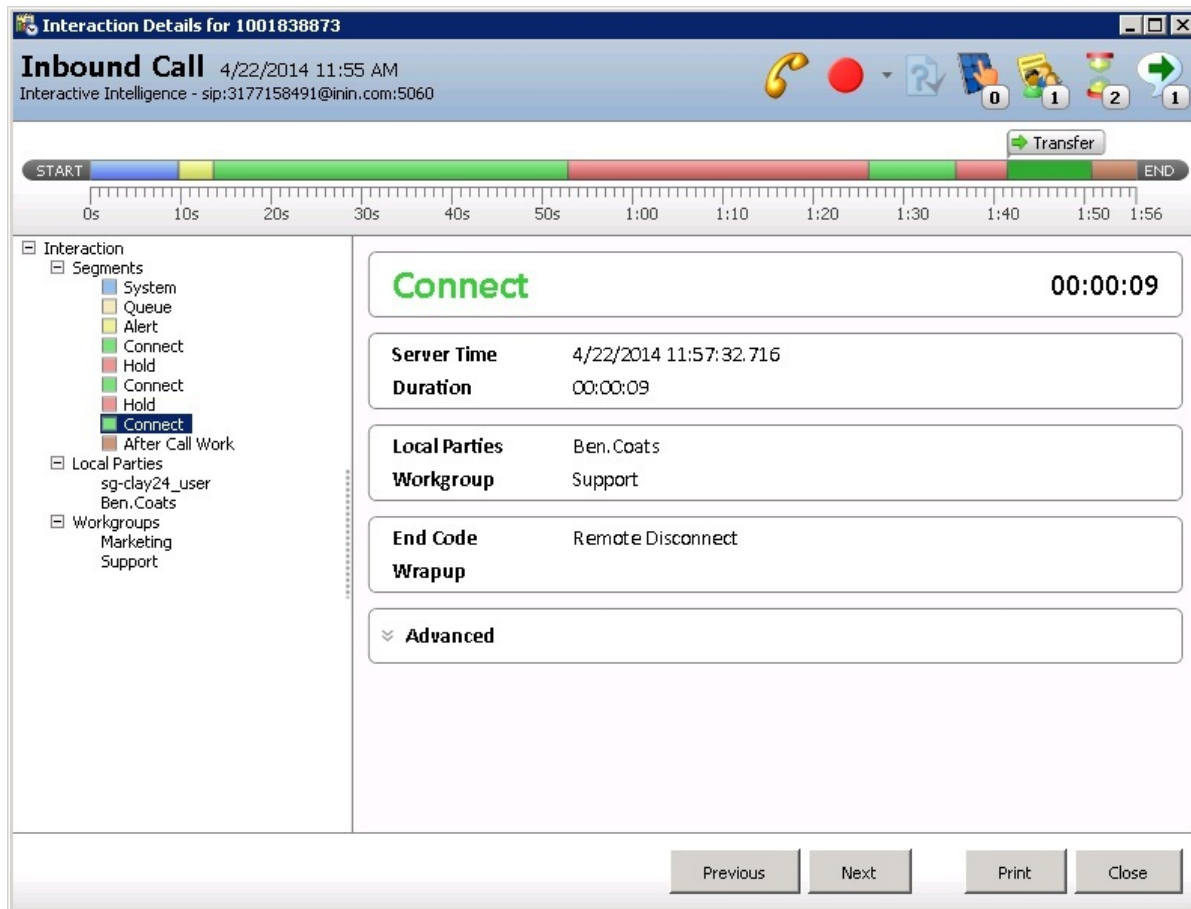
Consult transfer an ACD Interaction to another agent

Agent1 receives an ACD interaction from a remote party (for example, a customer). After interacting with this external party for a few minutes, Agent 1 put the call on hold. Agent 1 makes a consult call to Agent2. Agent1 and Agent2 consult for few minutes. Then Agent1 transfers the remote party call to Agent 2.

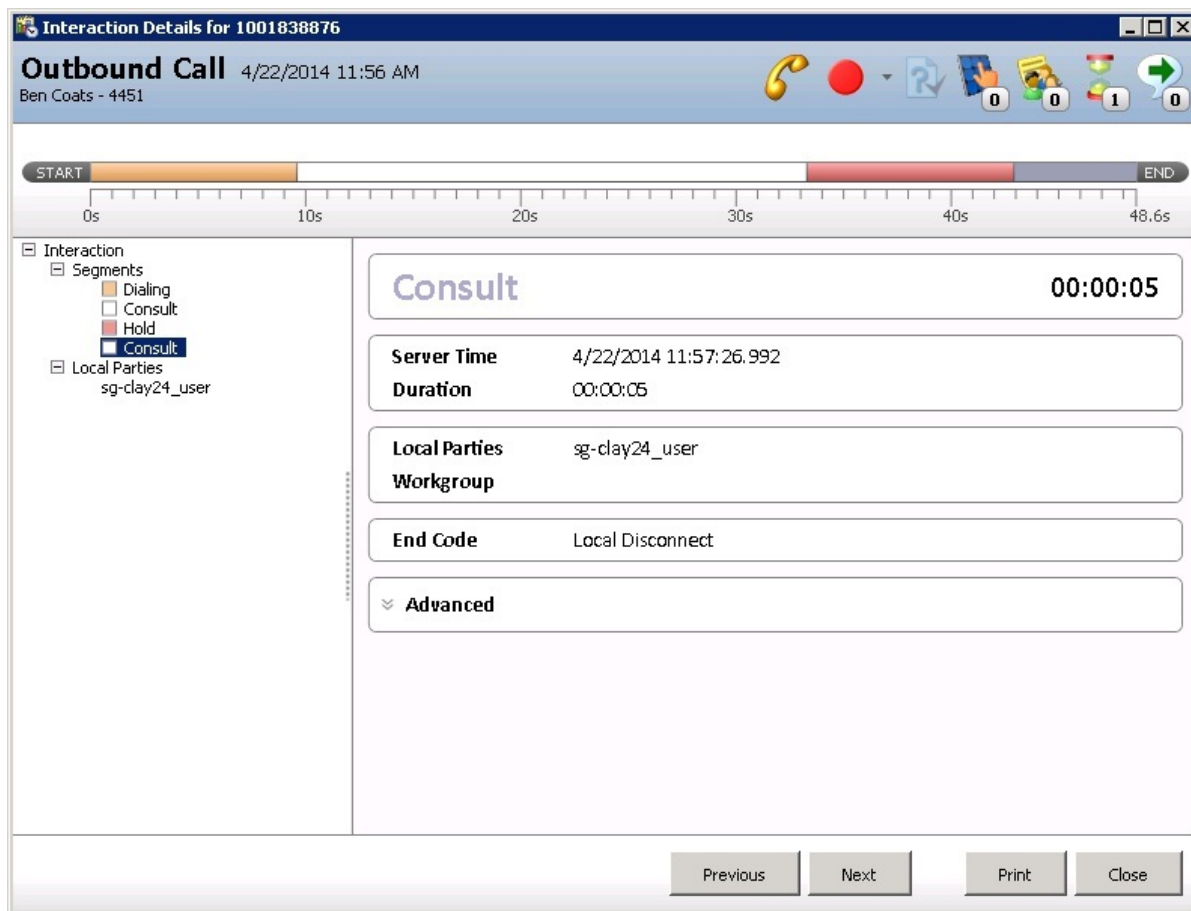
In this scenario, there are three InteractionIDs: inbound external call, initiator (Agent1) side of intercom consult call, and the respondent (Agent2) side of the intercom consult call. The external party InteractionID includes the pre-consult and post-consult transfer detail. The intercom call includes the consult call details.

Note: The consult interaction has the connect segment as Consult. This segment helps the Call Center manager identify this intercom interaction as a consulting interaction. In this example, Agent1 and Agent2 consult for some time, then Agent1 talks to the external party, and then Agent1 consults with Agent2 again. You see the corresponding held and consult segments in the intercom call when Agent1's context switches from the internal Agent2 to the external party.

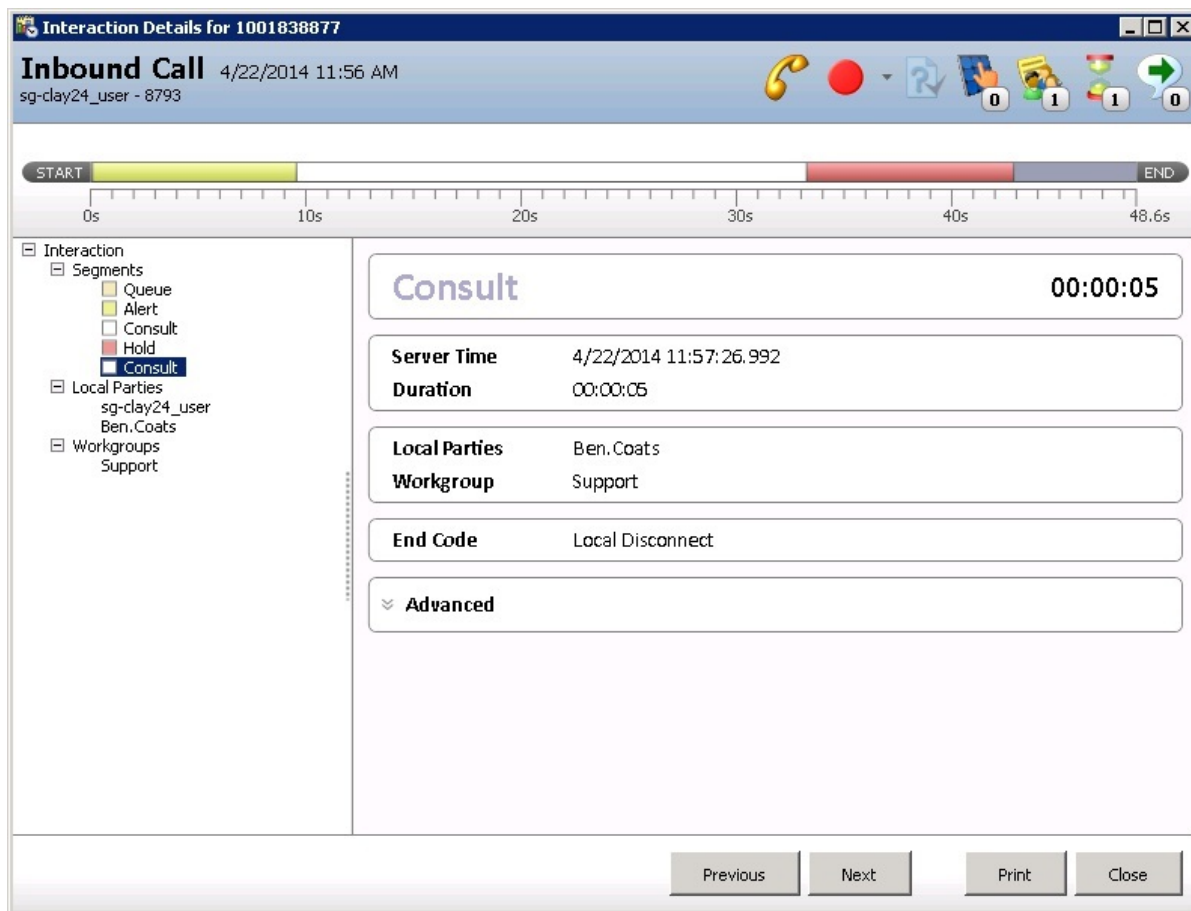
Inbound



Consult initiator



Consult respondent



Consult transfer an ACD call to a remote party

This scenario is similar to the [Consult transfer an ACD call to another agent](#) scenario. The difference is that Agent1 consults with a remote person instead of consulting an internal agent. Agent1 makes an outbound call instead of making an internal call. After consulting with the remote consulting party, Agent1 transfers the interaction.

Another difference is there are only two InteractionIDs instead of three InteractionIDs in this scenario. Because there is no intercom call, there are only two InteractionIDs: one for the inbound call and one for the outbound call. In the outbound call, notice the consult segment which represents the consulting operation.

Inbound

Interaction Details for 1001838891

Inbound Call

4/22/2014 2:28 PM

Interactive Intelligence - sip:3177158491@inin.com:5060

0 1 2 0

START

0s 10s 20s 30s 41.1s

END

Transfer

Interaction

Segments

System

Queue

Alert

Connect

Hold

Connect

Hold

External Transfer

Local Parties

sg-clay24_user

Workgroups

Marketing

Interaction Summary

Interaction ID

1001838891

Type

Call

Duration

00:00:41

Time

4/22/2014 2:28 PM

Disposition

Unknown

Direction

Inbound

Remote Address

sip:3177158491@inin.com:5060

DNIS

sip:3@sg-clay24:5060

Recorded

Yes

Surveyed

No

Remote Party

Interactive Intelligence

Local Parties

sg-clay24_user

Call Log

Previous

Next

Print

Close

Outbound

Interaction Details for 1001838894

Outbound Call
4/22/2014 2:28 PM
Indianapolis IN - (317) 222-2222

START
0s
10s
19.6s
END

Interaction
Segments
Dialing
Consult
Hold
Consult
Connect
Local Parties
sg-clay24_user

Interaction Summary

Interaction ID: 1001838894
Type: Call
Duration: 00:00:19
Time: 4/22/2014 2:28 PM
Disposition: Remote Disconnect during Connect

Direction: Outbound
Remote Address: +13172222222
DNIS:

Recorded: Yes
Surveyed: No

Remote Party: Indianapolis IN

Local Parties: sg-clay24_user

Call Log

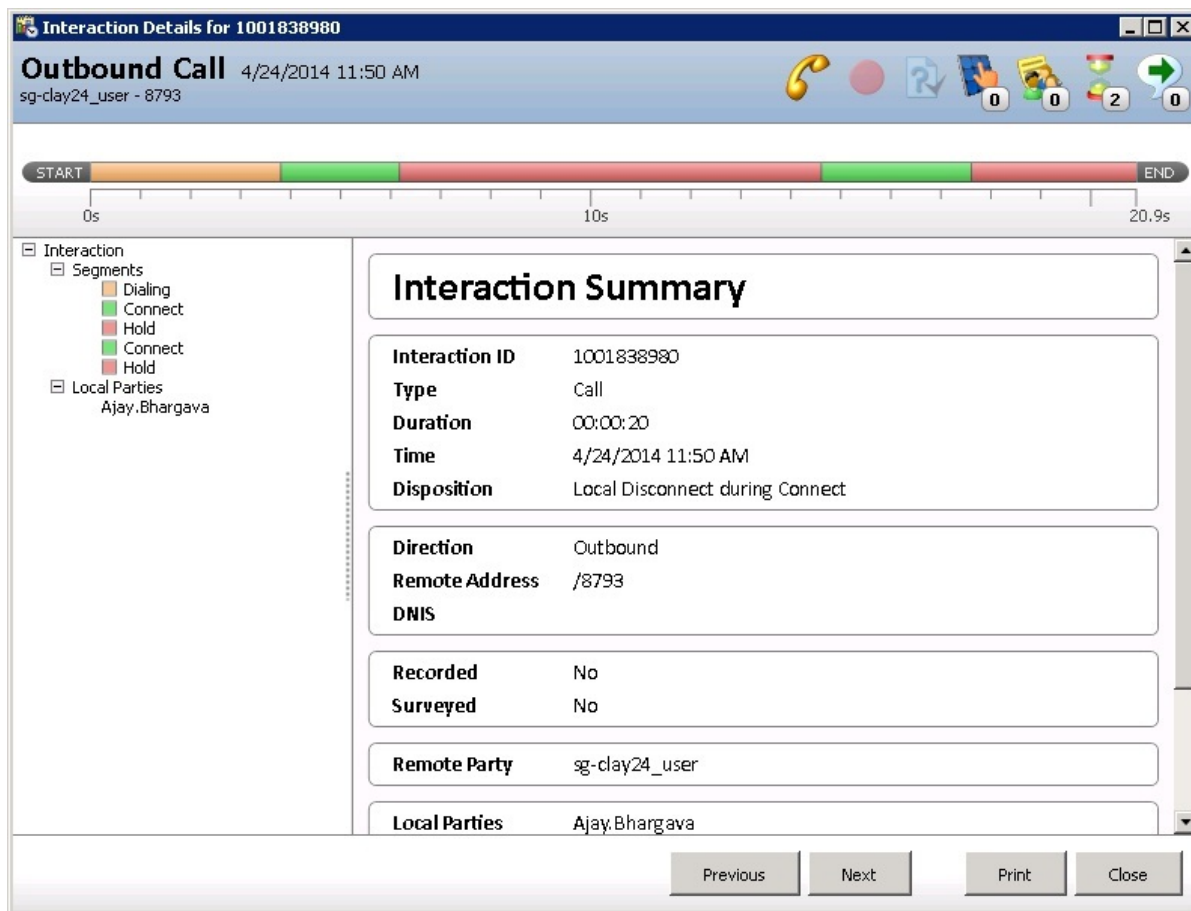
Previous
Next
Print
Close

Consult transfer an internal call to another agent

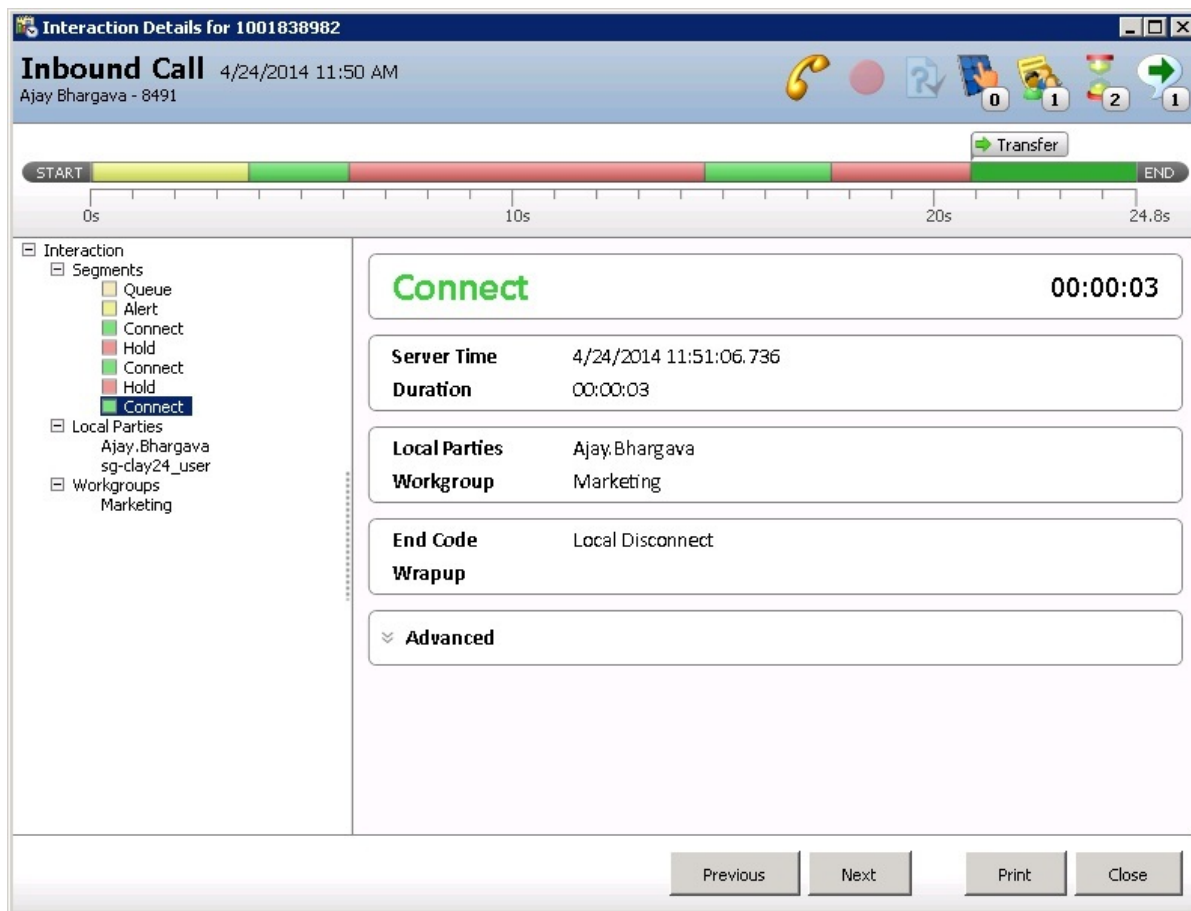
Only internal parties participate in this interaction. There is no external party. Agent1 makes an internal call to Agent2. Agent2 puts the call on hold. Agent2 consults Agent3 and then transfers the call from Agent1 to Agent 3. After the consult operation, Agent1 and Agent3 are connected.

Because this interaction is an intercom call, two InteractionIDs represent the initiator and respondent ends of the inbound call. For the consult call, there are two InteractionIDs again. So this scenario has a total of four InteractionIDs. The second set of InteractionIDs includes the Consult segment to represent the consult operation. The consult initiator and consult respondent InteractionIDs capture the consult duration. The system captures the connect duration after the consulting call in the intercom respondent and consult initiator InteractionIDs.

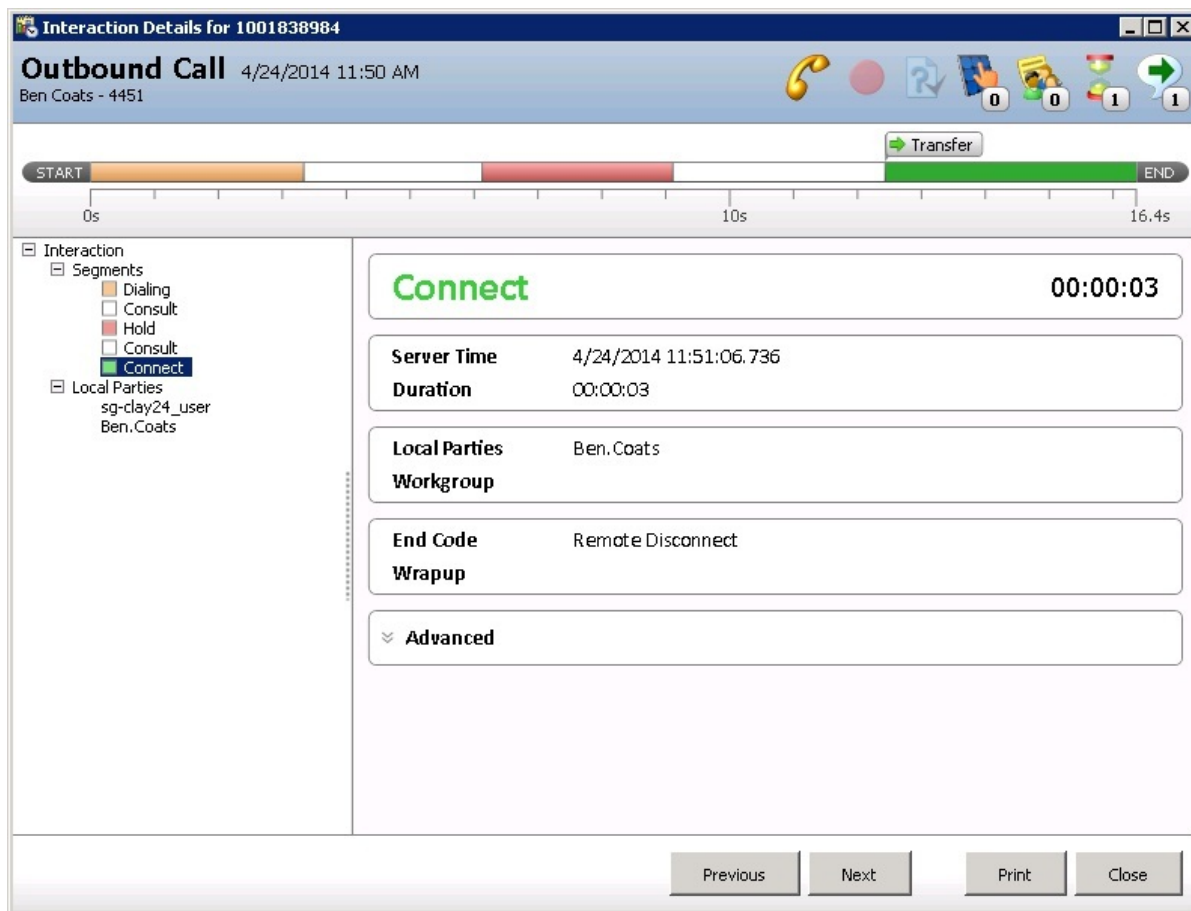
Intercom initiator



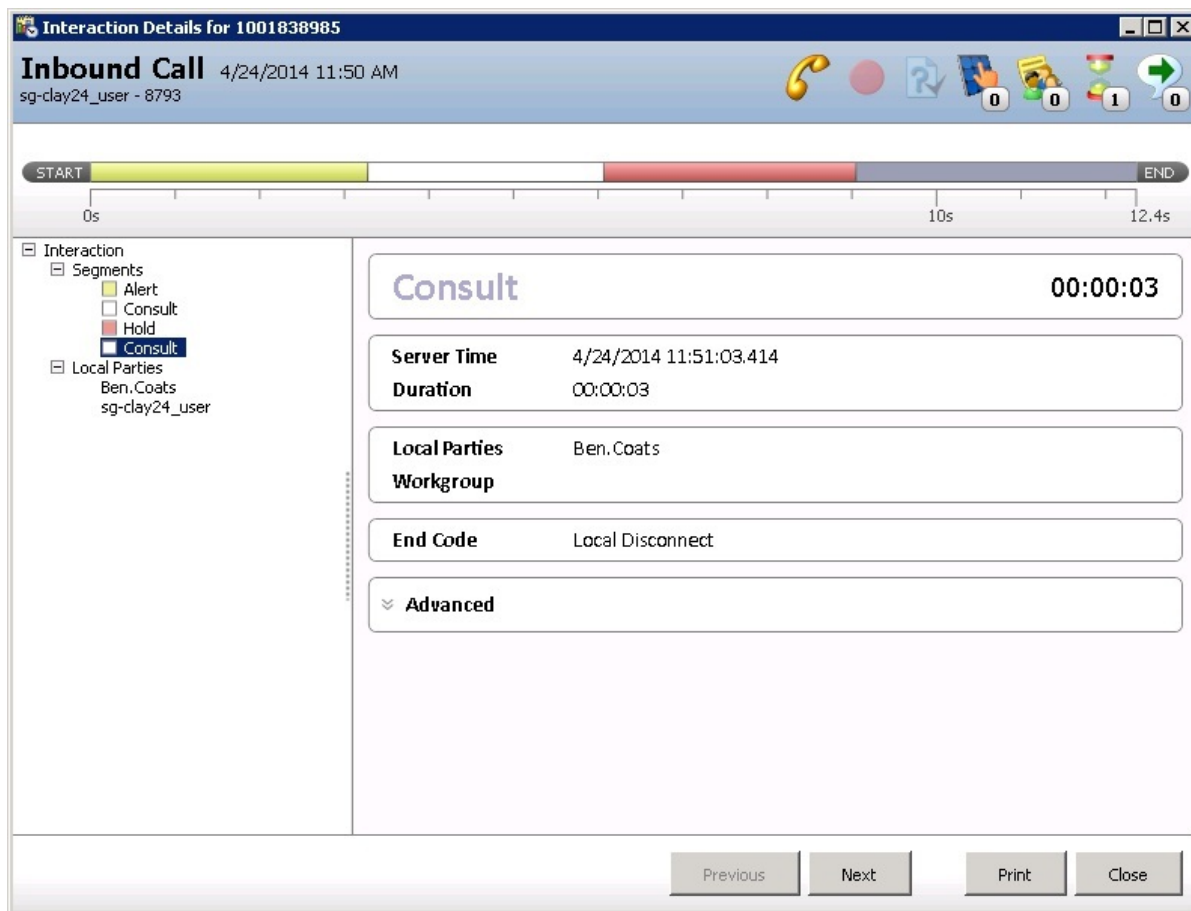
Intercom respondent



Consult initiator



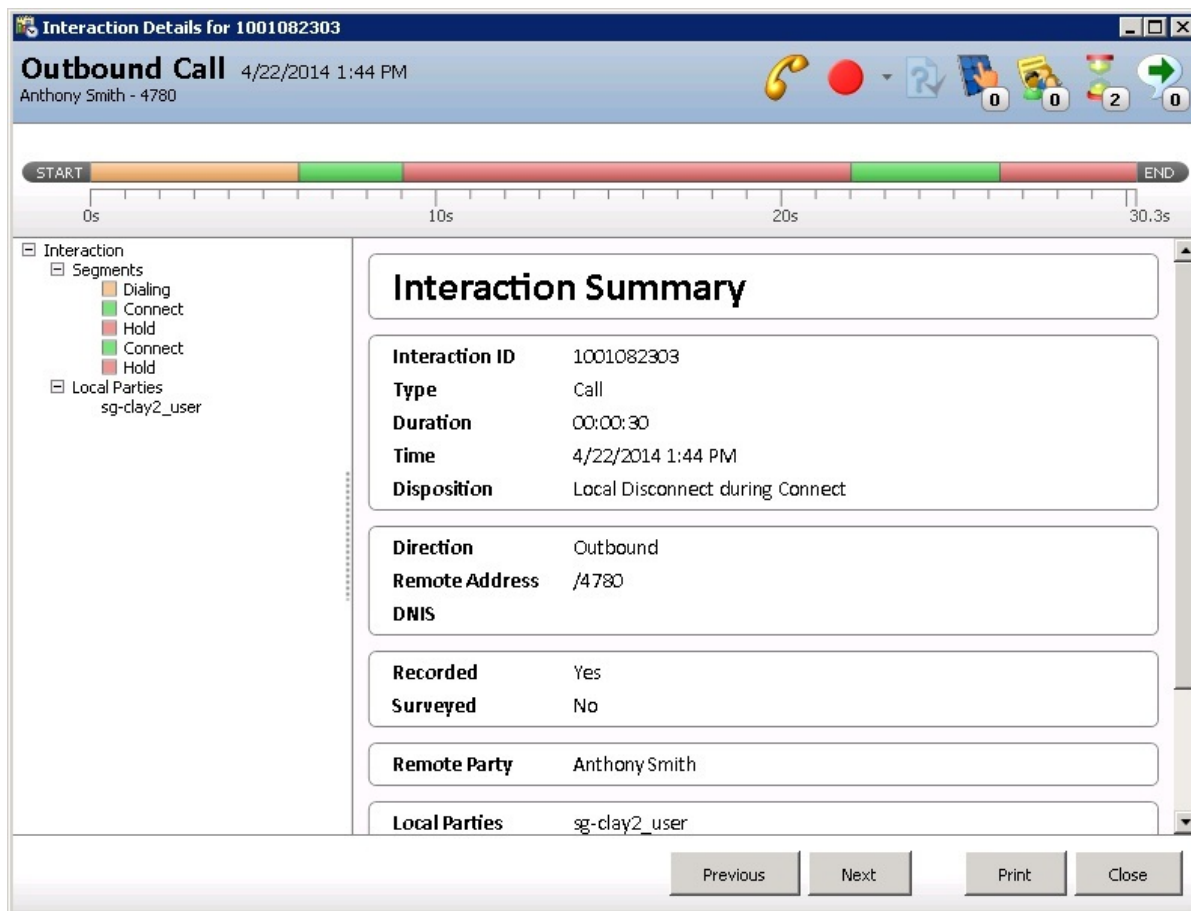
Consult respondent



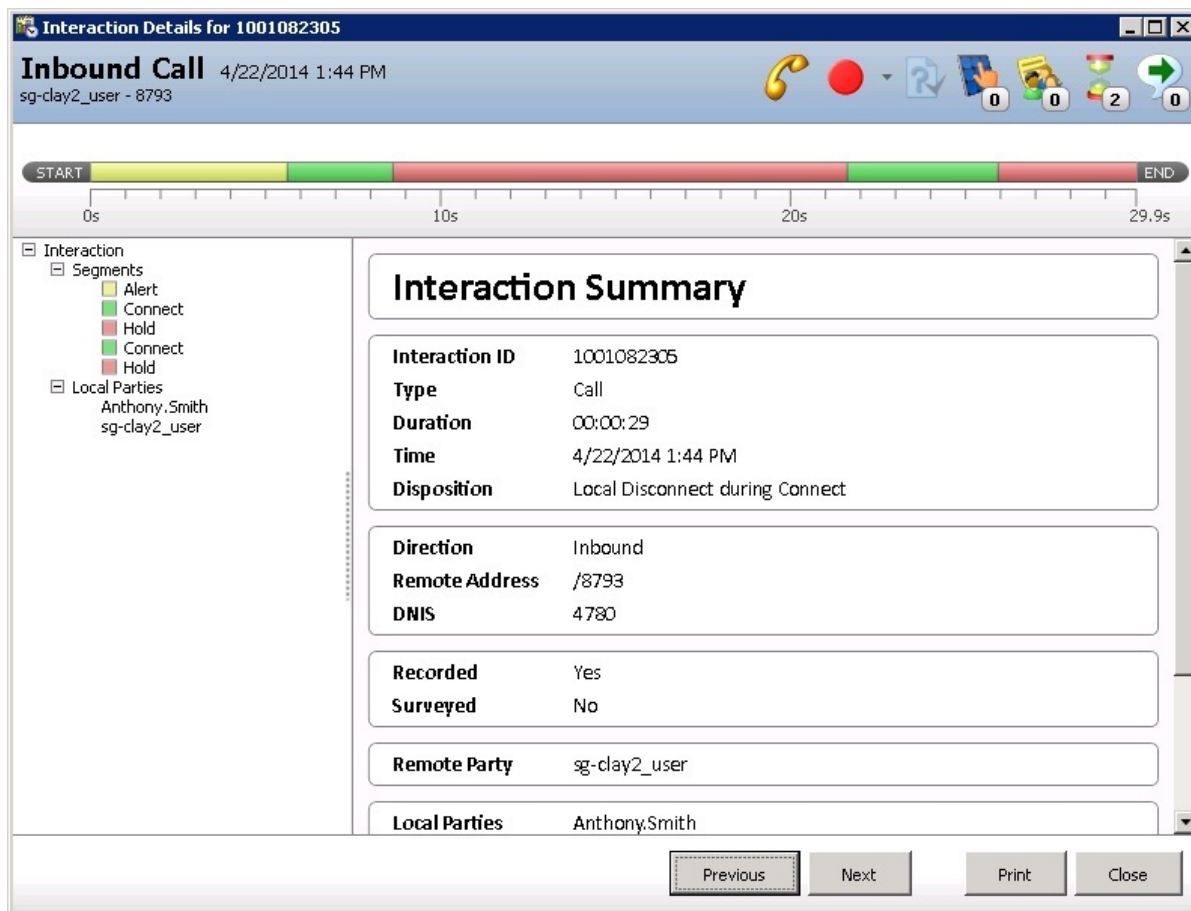
Consult transfer an internal call to a remote party

In this scenario, Agent1 calls Agent2. Agent2 puts the call on hold, and makes a consulting call by dialing an outbound number. After consulting with the remote party, Agent2 transfers Agent1 to the Remote Party. There are three InteractionIDs: the intercom initiator side, the intercom respondent side, and the outbound call (the consulting InteractionID).

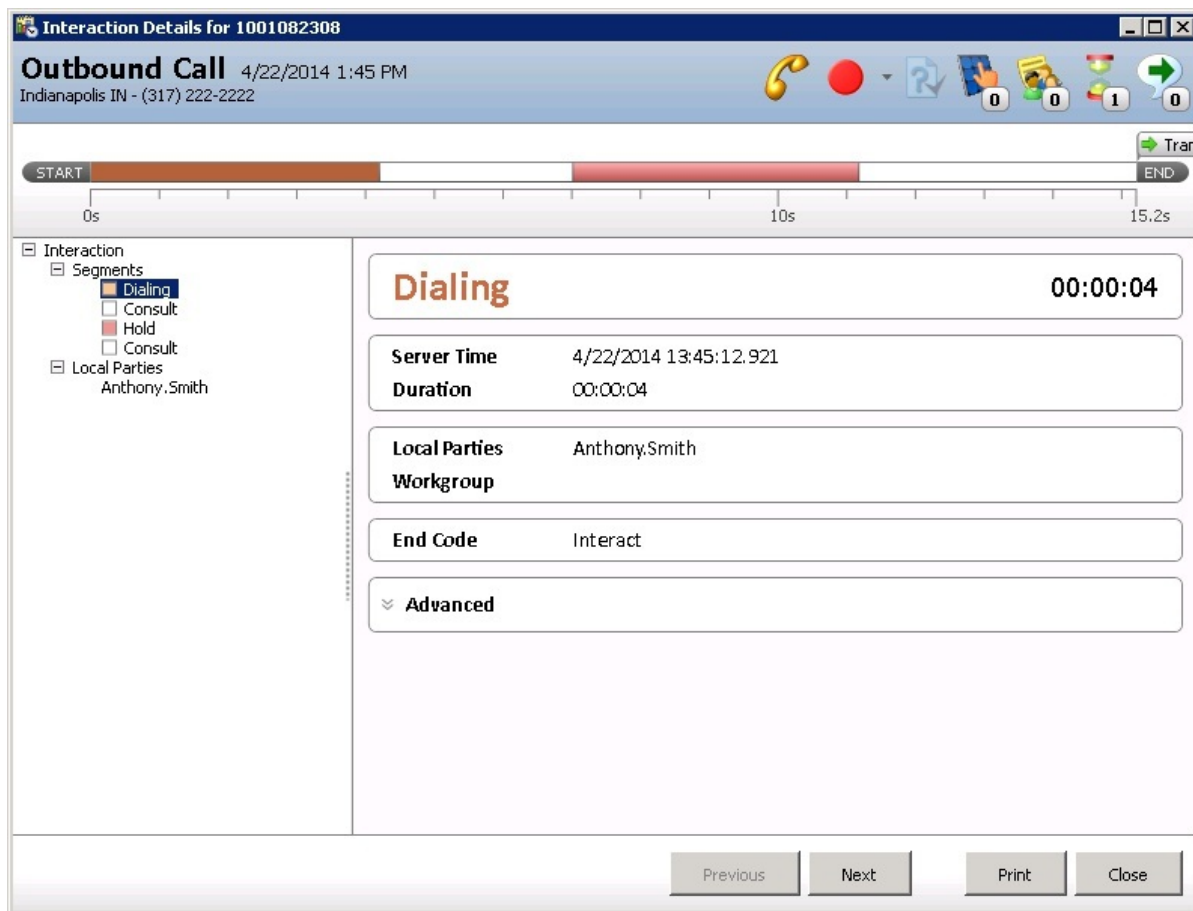
Intercom initiator



Intercom respondent



Outbound call



Three-party conference

Agent1 receives a call from an external party. Agent1 puts the call on hold and then calls Agent2. Agent1 and Agent2 consult, then Agent1 simply creates the conference with all the three parties,

There are three interaction IDs logged to the Interaction Segment Detail table:

- The Inbound ACD Call InteractionID captures all the events before the conference, in addition to conference detail.
- The intercom initiator InteractionID captures the name of the agent who handled the interaction before the conference and also the conference detail.
- The intercom respondent InteractionID represents the agent who consulted before the conference. It has the pre-conference segment and also the conference detail.

Refer to the conference segment advanced section detail. The local party contains all the associated conference participants.

Note: In this scenario, because both agents have a discussion before the conference, the intercom initiator and respondent InteractionIDs have a connect segment.

Inbound ACD call

Interaction Details for 3001805004

Inbound Call 11/13/2014 11:30 AM
Interactive Intelligence - (317) 715-8321

0s 1:40 2:57

START END

Interaction

- Segments
 - System
 - Queue
 - Alert
 - Connect
 - Hold
 - Conference
- Local Parties
 - Prabahar.Ignatius
 - sg-clay16_user
- Workgroups
 - Marketing

Workgroup sg-clay16_user

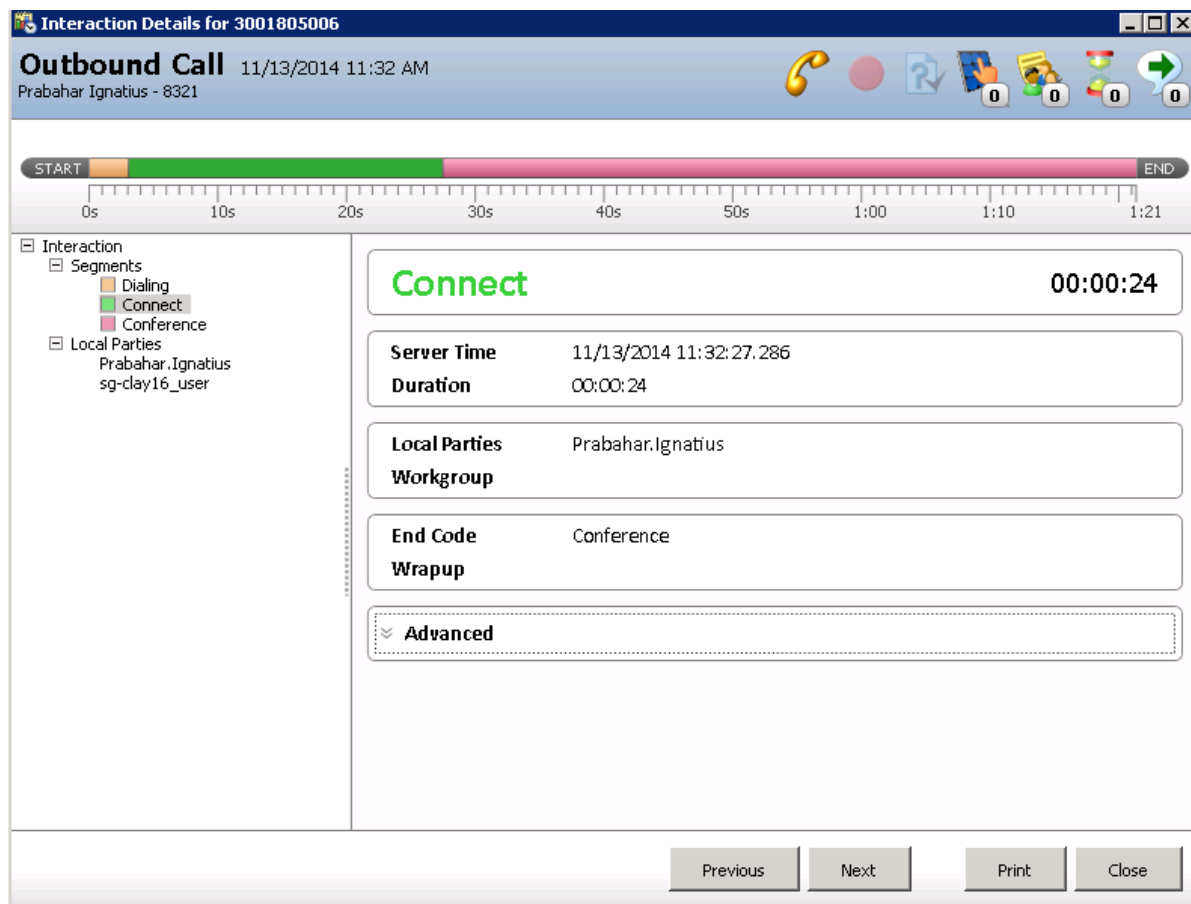
End Code Remote Disconnect

Advanced

Key	Value
InteractionKey	300180500450141113
SegmentSequence	0
Duration	53865
LocalParty	<ConfParties> <ConfParty Role="Initiator" External="F" Name="Prabahar Ignatius" /> <ConfParty Role="Respondent" External="F" Name="sg-clay16_user" /> <ConfParty Role="Participant" External="T" Name="Interactive Intelligence" /> </ConfParties>
Queue	
EndCode	RemoteDisconnect
Details	
ClientUtc	20141113T113646.834-05
SegmentSequence	0

Previous Next Print Close

Intercom initiator



Intercom respondent

Interaction Details for 3001805007

Inbound Call
11/13/2014 11:32 AM
sg-clay16_user - 8793

START
END
0s 10s 20s 30s 40s 50s 1:01

Interaction

Segments

Alert
Connect
Conference

Local Parties
Prabahar.Ignatius
sg-clay16_user

Time
11/13/2014 11:32 AM
Disposition
Unknown

Direction
Inbound
Remote Address
/8321
DNIS
8793

Recorded
No
Surveyed
No

Remote Party
sg-clay16_user

Local Parties
Prabahar.Ignatius
sg-clay16_user

Call Log

Advanced

Previous

Next

Print

Close

Outbound Interaction

Dial on behalf of workgroup outbound interaction

We capture all the required segments of a simple outbound interaction. In this scenario, the outbound interaction is associated with a workgroup. This scenario is similar to the Dialer-initiated outbound interaction. However, Dialer-initiated outbound interactions are not associated with a workgroup until the remote party picks up the call. When this interaction is assigned to an agent, it is associated with a workgroup.

But an agent can associate a workgroup with an interaction when placing a call. All the segments in these *Dial on behalf of* interactions include the workgroup association. This workgroup association appears in the first segment (Dialing) and in every other segment until the interaction ends.

Interaction Details for 3001805063

Outbound Call 11/16/2014 7:29 PM
Indianapolis IN - (317) 222-2222

START 0s 10s 15s END

Interaction
Segments
Dialing
Connect
Hold
Connect
Local Parties
Prabahar.Ignatius
Workgroups
Marketing

Dialing 00:00:04

Server Time 11/16/2014 19:29:13.384
Duration 00:00:04

Local Parties Prabahar.Ignatius
Workgroup Marketing

End Code Interact

Advanced

Previous Next Print Close

Simple outbound interaction

In this case, the agent made an outbound call to a remote number. Since the agent did not associate this call with a workgroup, you do not see a workgroup name in any of the segments.

Interaction Details for 3001805067

Outbound Call

11/16/2014 7:29 PM

Indianapolis IN - (317) 222-2222



START

0s

Dialing at 00:00:00

10s

12.3s

END

Interaction

Segments

Dialing

Connect

Hold

Connect

Local Parties

Prabahaar.Ignatius

Dialing

00:00:03

Server Time

11/16/2014 19:29:46.154

Duration

00:00:03

Local Parties

Prabahaar.Ignatius

Workgroup

End Code

Interact

Advanced

Previous

Next

Print

Close

Interaction Detail API

Use the Interaction Detail API to access and manipulate interaction segment detail data.

[Table 0 - Events](#)

[Table 1 - Attributes](#)

[Table 2 - Attribute detail data](#)

[Table 3 - Errors](#)

[Sample code](#)

Table 0 - Events

Column	Description
Interaction Id	Value from InteractionSegmentDetail.InteractionIdKey
EventId	Integer value for event name in "Name" column
EventKey	Unique Id generated by InteractionDetailsAPI
ServerUTC	Server time when the record was entered
Server Seq	Sequence
ClientUTC	Client time when the record was entered
Name	Localized event name
NameResourceString	Segment Detail with tokens
FinalResourceString	Segment detail with replacement strings

Table 1 - Attributes

Column	Description
AttributeKey	Unique Id generated by InteractionDetailsAPI
EventKey	Unique Id generated by InteractionDetailsAPI; foreign key to Events table
SeqNum	Sequence
Name	Abbreviated description of attribute
Type	Attribute data type
Action	Specific action of attribute (eg., 'DELAY', 'ALERT', 'CONNECT')
AttrNameResourceString	Name of value in 'DisplayValue' column
DisplayValue	Value of attribute

Table 2 - Attribute detail data

Column	Description
EventDetailKey	Unique Id generated by InteractionDetailsAPI
EventKey	Unique Id generated by InteractionDetailsAPI; foreign key to Events table
Action	Specific action of attribute (rg., 'DELAY', 'ALERT', 'CONNECT')
Description	Event description
DurationSeconds	Duration of event in seconds, up to 3 decimal places (thousandths); eg: '14.103', '120.502'
UserQueue	User queue associated with event; blank if none
StationQueue	Station queue associated with event; blank if none
WorkgroupQueue	Workgroup queue associated with event; blank if none
EndCode	End code associated with event
InteractionCount	Place holder for analytics package to add '1' or '0' to get accurate interaction count by various dimensions
Connected	'True' of 'False', depending if 'CONNECTED' appears in the attribute list of the given event.
SegmentSequence	The sequence number of the segment; canbe used for sorting by segment order

Table 3 - Errors

Column	Description
ErrorText	The text of the errors encountered from (1) while translating URI value to XML, (2) invalid dates, (3) attributes with invalid replacement tokens in Events.NameResourceString, (4) invalid XML returned from QiHistory.MultipleEventsToXML

Sample code

The following code sample uses `ININ.Reporting.InteractionDetailsAPI.dll`. You can find this, along with `ININ.Reporting.Historical.InteractionDetailsAPI.dll` and `QiHistoryInterfaceLib.dll`, in your IC Business Manager Apps directory. This directory is typically `C:\Program Files (x86)\Interactive Intelligence\BusinessManagerApps`.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using System.Data.SqlClient;
using ININ.Reporting.Historical.InteractionDetailsAPI;
using QiHistoryInterfaceLib;
namespace InteractionDetailExample
{
    class Program
    {
        public static readonly string RAW_DATA_SOURCE = @"Data Source=YourSQLServerHere;Initial
Catalog=YourDatabaseHere;User Id=YourUserIdHere;password=YourPasswordHere;timeout=600";

        static void Main(string[] args)
        {
            DataSet dsRawInteractionDetailRecords = GetInteractionsDataSet();

            //Example 1:
            // This method will result in all the data queried from InteractionSegmentDetail be put into
            memory; be cautious of this method if you're
```

```

        // querying a large quantity of data
        DataSet ds = ConvertSegmentDetailToDataSet(dsRawInteractionDetailRecords);
        //continue here with using the resulting tables as necessary...
        //Example 2:
        // This method will return an IDataReader, which means that you can iterate through it, as in the
example code, without
        // loading the entire DataSet into memory at once.
        // Note the call to QiHistory.MultipleEventsToXML; this converts the string you see in
InteractionSegmentDetail.SegmentLog to XML.
        // The call to InteractionDetailsAPI.AddXMLData converts the XML to an array of 4 .Net
System.Data.DataTable objects.
        IDataReader dr = ConvertSegmentDetailToDataReader(dsRawInteractionDetailRecords);
        while (dr.Read())
        {
            string EventLogXML = QiHistory.MultipleEventsToXML(dr["SegmentLog"].ToString(), false,
false, QiHistory.eSchemaType.Localized);
            DataTable[] dtDetails = InteractionDetailsAPI.AddXMLData(dr["interactionidkey"].ToString(),
EventLogXML);
            //continue here with using the resulting tables as necessary...
        }
        dr.Close();
    }
    static DataSet ConvertSegmentDetailToDataSet(DataSet dsDetailRecords)
    {
        InteractionDetailsAPI.ReportProgress += new EventHandler<IntEventArgs>
(InteractionDetailsAPI_ReportProgress);
        DataSet dsConvertedDetailRecords = InteractionDetailsAPI.GetAsDataSet(dsDetailRecords.Tables[0],
QiHistory.eSchemaType.Localized);
        return dsConvertedDetailRecords;
    }
    static void InteractionDetailsAPI_ReportProgress(object sender, IntEventArgs e)
    {
        //report progress here if required
    }
    static IDataReader ConvertSegmentDetailToDataReader(DataSet dsDetailRecords)
    {
        IDataReader drSource = GetInteractionsDataReader();
        return drSource;
    }
    #region Utilities
    /// <summary>
    /// Gets some data from InteractionSegmentDetail
    /// </summary>
    /// <returns>A complete DataSet</returns>
    static DataSet GetInteractionsDataSet()
    {
        DataSet dsRet = new DataSet();
        using (SqlConnection dbConnection = new SqlConnection(RAW_DATA_SOURCE))
        {
            string SQL = "SELECT top 10 InteractionIDKey, SegmentLog FROM InteractionSegmentDetail WHERE
((StartDateTimeUTC >= cast('2014-05-27 04:00:00.000' as DateTime)) AND (StartDateTimeUTC <= cast('2014-05-27
13:59:59.000' as DateTime)))";
            using (SqlCommand dbCommand = new SqlCommand(SQL, dbConnection))
            {
                using (SqlDataAdapter dbAdapter = new SqlDataAdapter(dbCommand))
                {
                    dbAdapter.Fill(dsRet, "Test");
                    return dsRet;
                }
            }
        }
    }
    /// <summary>
    /// Gets some data from InteractionSegmentDetail
    /// </summary>
    /// <returns>an IDataReader for less memory intense processing</returns>
    static IDataReader GetInteractionsDataReader()
    {
        string SQL = "SELECT top 10 InteractionIDKey, SegmentLog FROM InteractionSegmentDetail WHERE
((StartDateTimeUTC >= cast('2014-05-27 05:00:00.000' as DateTime)) AND (StartDateTimeUTC <= cast('2014-05-27

```

```
13:59:59.000' as DateTime)))";  
    SqlConnection conn = new SqlConnection(RAW_DATA_SOURCE);  
    SqlCommand cmd = new SqlCommand(SQL, conn);  
    conn.Open();  
    SqlDataReader dr = cmd.ExecuteReader();  
    return dr;  
}  
#endregion  
}
```

Miscellaneous

Snoozed callback and interaction details

Examples of how snoozing a callback interaction affects Interaction Details are available in this KB article: https://genesyspartner.force.com/customercare/pkb_Home?id=kA50B0000008R1ZSAU&l=en_US&fs=Search&pn=1

Switchover

In switchover-configured CIC servers, when the interaction reliability feature is enabled for email and chat interactions, we might log two records for the active interactions at the time of switchover in the Interaction Summary and Interaction Segment detail tables. When these interactions are logged with same Interaction ID key, the sequence number is incremented. Always use the interaction with higher value sequence number.

Here is how active interactions are processed by the CIC subsystem, Tracker Server, at the time of switchover or shutdown.

In the event of an IC shutdown with active interactions, Tracker Server attempts to salvage interaction segment data by writing into PMQ files in the PMQSegmentLogs directory. On the next restart, Tracker Server picks up those files and persists data in the database.

In the same way, Tracker Server also attempts to salvage interaction segment data during a switchover. Tracker Server attempts to capture as much interaction information as reliably as possible. Since shutdown or switchover leaves the system in a flux state, Tracker Server uses default values for active interactions during shutdown. If real time values can't be determined, Tracker Server uses default values. For example, if segment data is flushed due to shutdown, Tracker Server uses the string 'SHUTDOWN' as default string data. For switchover, Tracker Server uses 'SWITCHOVER' as default string data. For integer values, Tracker Server uses '0' as the default value. For timestamp fields, it uses an unrealistic default value like '1969-12-31 19:00:00.000' or '1970-01-01 00:00:00.000(GMT).'

In the InteractionSummary table CallEventLog field, Tracker Server always appends a string at the end. This indicates if the data record was collected during shutdown or switchover, along with the timestamp indicating when the data record was dumped. It is approximately the same as the shutdown or switchover time.

```
09:42:24: Offering 09:42:24: Sent to user user1 09:42:24:Initializing
09:42:23: Sent to user user2 09:42:24: Internal Call:user1
09:42:24: Dialing 09:42:24: User Alert:user1 09:42:25: Alerting
09:42:25: Sent to station woprjr1 09:42:25: Connected 09:42:39: Disconnected
[Remote Disconnect] 09:42:51: SHUTDOWN
```

In regular cases (not a shutdown or switchover), there is no such message. Administrators can use this field to determine if the data was collected during shutdown or switchover and be advised that data may not be completely correct. Tracker Server has done only a best possible attempt to salvage tracking data on active interactions.

Multiple ACW tasks

In your Call Center, if agents are going to do multiple After Call Work tasks after an ACD interaction, set these two server parameters to **true**. In this way, multiple After Call Work (ACW) segments are captured successfully in interaction Segment detail log.

Server Parameter Name	Value
StatServer_AllowMultipleACW	true
StatServer_AlwaysTrackACWForLastACDInteraction	true

Change Log

Date	Changes
09-January-2012	Updated Copyright and Trademark Information page.
13-March-2015	- Page 2: Segment detail data documentation. Renamed section and removed 2 bullet points. - Pages 32-34: Interaction Detail API. Added new section. - Page 35: Snoozed callback and interaction details. Added new section. Converted bullet point from page 2 to a paragraph and moved it. - Made changes to client for transition to Interaction Desktop. Added CIC client section.
31-March-2015	Corrected typo on page 6.
28-July-2015	- CIC client - Removed reference to Interaction Client .NET Edition. - Removed from boilerplate: Starting with CIC 2015 R3, Interaction Desktop replaces Interaction Client .NET Edition as the primary CIC client. - Schema and table detail - Verified name of CIC Data Dictionary Technical Reference in CIC 2015 R3 Doc Library. Segment detail data in different interaction scenarios - Verified screen captures against 2016 R1 server.
16-February-2016	- In the "How CIC generates segments" section, added "Park" to the "Segment Type" table. - In the "Segment Details: End Codes" section, added "Park" to the "End Code" table. - Added the "Park scenario" section. - Updated Copyright and Trademark Information page.
12-January-2017	Converted to HTML format.
25-April-2017	Updated documentation to reflect the removal of Interaction Client Web Edition.
05-May-2019	Reorganized the content only, which included combining some topics and deleting others that just had an introductory sentence such as, "In this section..." For more details, see https://devjira.inin.com/browse/CICDOC-180.