

Genesys_T-Server-UCCE_Integration_Problem_Isolation

The following is intended to help end users isolate UCCE/T-Server integration problems and to determine next steps in identifying solutions. It supplements, and is a prerequisite to, the information on troubleshooting specific Cisco Genesys T-Server/PG integration issues at

http://docwiki.cisco.com/wiki/Troubleshooting_Tips_for_Unified_CCE_8.0#Genesys_T-Server.2FSystem_PG_integration

This page and the troubleshooting Wiki cover some common debugging issues, but are not intended to be an exhaustive Genesys T-Server or URS troubleshooting guide. They are intended to help narrow down apparent system issues and guide the reader to the most appropriate place to look next, whether it be configuration, scripting, or a more serious issue that would need to have a case opened or further troubleshooting performed.

You should already be at least familiar with troubleshooting of UCCE systems. In addition, familiarity with Parent/Child and the components of the Genesys System in use (whether it be T-Server and desktop only, or a complete Genesys Enterprise routing system with URS, Stat Server, and so on) is helpful) You should also be familiar with CTI Server logs, particularly in a parent/child deployment. Generally it should not be necessary very often to look beyond CTI Server to determine if the next logical place to look at a problem is down in the UCCE System, or up in the Genesys Enterprise system (or simply in the T-Server/desktop for a CTI-only deployment).

Contents

- 1 Configurations
 - ◆ 1.1 Genesys CTI-Only
 - ◆ 1.2 Genesys in the Enterprise
 - ◆ 1.3 Genesys Only Call Variables
- 2 Important Best practices
- 3 Basic functionality Tests - Prerequisites
 - ◆ 3.1 UCCE Side - No T-Server Connected
 - ◆ 3.2 Genesys Side (Enterprise Routing)
- 4 Common Areas of Issues ? Things to Double Check
 - ◆ 4.1 Agent password synchronization
- 5 Diagnosing Softphone Issues
 - ◆ 5.1 Basic Behavior Test when Using T-Server/Genesys Desktop
 - ◆ 5.2 CTI vs Call Setup Issues
- 6 Important Information/Special information
 - ◆ 6.1 Trace Mask Level
 - ◆ 6.2 Useful CTI Server procmon Commands
 - ◇ 6.2.1 list_monitors (lm)
 - ◇ 6.2.2 dump_device (dd)
 - ◇ 6.2.3 list_clients (clients)
 - ◇ 6.2.4 list_routing_devices (lrd)
 - ◆ 6.3 Using CTI-TEST all events mode to compare
 - ◆ 6.4 PERIPHERAL MONITOR mode

Configurations

With all the possible configurations there are really only two we need to distinguish between: Genesys CTI-Only, and Genesys in the enterprise. These are described briefly below for identification. Also, there are slight differences between them. For practical purposes Genesys CTI-Only is basically a UCCE integration with a third party (or custom) CTI desktop and server application. With one notable difference, the Genesys Enterprise is virtually identical to a UCCE/ICM parent/child deployment.

Genesys CTI-Only

In the Genesys CTI-Only model one or more T-Servers and groups of Genesys desktop software are connected to one or more System PG's on one more more UCCE systems (The System PGs may be on the same UCCE system or different systems). Note that one T-Server (or Primary/Standby T-Server Pair) is required for each System PG on the UCCE System(s).

Genesys in the Enterprise

In this model, each UCCE controlled System PG/CG is seen as an ACD from the point of view of Genesys. Each link to each System PG is seen as a separate ACD to Genesys. Genesys may control other non UCCE ACD's as well as network VRUs, etc. Genesys may queue in the network (Via GVP) and route to either UCCE Call Type/Script/Skill group associated route points, directly to agents, or to 'remote' route points and then to agents.

In this case this is analogous to an ICM system with PGs over many ACDs except that in this case it's Genesys over many ACDs which happen to be UCCE System PG deployments.

Agents whose skill group will be targeted by Genesys must not be directly targeted by UCCE. For example, if Agent 1000 is in Skill Group 1 who will be targeted by Genesys (through a TRP the Route point running the script to queue/distribute calls is controlled by UCCE), Genesys must only target that agent through the route point and not directly. (Exceptions can be made for back up standby scripting). It is acceptable in failover situations for agents to be targeted briefly by both Genesys and the UCCE's router. At the time of this writing each site was limited to being either ALL Genesys targeted agents, or ALL UCCE targeted agents (Genesys would target UCCE controlled route points).

All agents must be members of non-default skill groups. An example of Standby would be an agent that normally is directly targeted by Genesys, but if they are not connected or the connection drops the agent's skill group can have calls distributed to it by UCCE.

Worth noting in this deployment is the UCCE PGs are the ACD so expect references to that from the customer/end/Genesys documentation user referring to UCCE as the ACD.

Genesys Only Call Variables

Depending upon how the system is deployed it is possible that Genesys only call variables exist in the T-Server on up (Genesys Desktop, URS, etc.). Many variables can be defined in Genesys, and depending upon the mapping of the variables done in the T-Server configuration some of these may never be seen on UCCE (or UCCE desktops).

In a deployment where calls between two System PGs on the same UCCE system are translation routed by UCCE, the T-Servers can still track the calls in some circumstances via the RouterCallKeyDay and RouterCallKeyCallID provided by UCCE. This only works for calls that run a local script and are translation routed to the destination PG. This can allow the Genesys only call variables to survive a PG to PG translation route even though Genesys is not doing the routing and UCCE is unaware of these variables.

Important Best practices

Have backup scripts for ALL remote route points/DNs

All 'remote' route points, whether used for call matching TRPs or for initiating Genesys controlled consults should have backup scripts in UCCE. Any DN on UCCE where 'Permit Application Routing' is checked

should have backup UCCE scripts.

There should be a call type associated with the route point DN, a script, and default routing logic. If the link between CTI Server and the T-Server goes down this logic can/will take over and use UCCE to route any calls.

It is suggested that agents that are normally directly targeted by Genesys have at least one specific skill group they are a member of that is ONLY targeted by UCCE. This will allow tracking/reporting of the calls to tag them as ?calls handled while in standby?; i.e., if skill groups 4, 5, and 6 only get targeted by standby scripts, any calls to them can be accounted for in time where the link was down or some issue had occurred.

Additionally, the UCCE script administrator may want to use unique call types for these backup call types/scripts to allow easy identification of them via TCD or call type reporting if desired.

Why bother setting up a backup configuration first?

- Having this backup configuration allows for proper pre-testing of an entire UCCE configuration.
- If you have backup scripts for all DNs, even remote DNs you can run local UCCE call scenarios that will ensure proper operation and configuration from the call manager on up. Possibly this can even test gateways if you have a loopback number available.
- If you don't do this and there is a routing issue you'll (potentially) be looking at jtapigw, PIM, OPC logs, etc.. If you do this and all works most likely you'll only have to look at the Permit Application Routing flag and CTI-Server logs to see if UCCE is performing properly.

Basic functionality Tests - Prerequisites

UCCE Side - No T-Server Connected

In deployments where Genesys is doing the enterprise routing, prior to hooking up the T-Server, it is important to confirm all proper behavior of the UCCE system to make problem area diagnosis easier. So with the T-server not connected try and ensure the following work.

- Make agents available and try basic call flows for all ICM controlled route points. Call the route point and make sure the calls get distributed to the agents, etc.. Conclusion: All CM route points for these calls are set up correctly: DNs, scripts, etc. are set up correctly.
- For all of the normally Genesys controlled route points confirm the same. Make calls to the Route points from a phone and ensure the backup routing scripts are in place and route the calls to available agents. Conclusion: The CM and ICM setup aside from checking ?PermitApplicationRouting? on the Dialed Number table can be confirmed as correct.
- Check that RONA works for all ICM controlled routing ? Checking one call per Agent Desk Setting is probably more than sufficient.

Genesys Side (Enterprise Routing)

When Genesys is doing the enterprise routing, check everything that can be checked without sending the call to UCCE (Scripting, prompting, call treatment, and so on) prior to adding the scripting/configuration necessary to send the calls to UCCE.

Common Areas of Issues ? Things to Double Check

- Ensure that all route points that Genesys will be using as TRPs are configured on the call manager, in the UCCE Dialed Number table and that they have ?Permit ApplicationRouting? enabled.
- Ensure each T-Server for a given System PG has both the A and B side IP addresses configured (If applicable) and the proper ports and peripheral ID?s.
- Ensure the Genesys Call variable mapping is in place and proper.
- Ensure the T-Server(s) is/are connecting properly.
- Ensure all the necessary agent devices and route points are configured in Genesys. The T-Servers use Peripheral Monitor mode and if not configured, no events will be received for a device. Agent instruments and route points should be monitored.
- Ensure that your UCCE system is 8.0(2) or later.
- Ensure the passwords are synchronized (Set the passwords the same on both UCCE and Genesys ? See below).

Agent password synchronization

One anticipated area of confusion/trouble is password synchronization. Agents configured in the Genesys domain have passwords, and agents configured in the UCCE domain have passwords. **It is imperative that these match.** If these don?t match, agent login from the Genesys side will fail. If an agent fails a limited number of times they will be locked out from attempting to log-in again for some time period (possibly 60 minutes).

This is not an automated process. Set the agent password first on Genesys, then set the password on the UCCE side to the identical value.

Diagnosing Softphone Issues

Basic Behavior Test when Using T-Server/Genesys Desktop

If during any call scenario undesired behavior is observed on the Genesys desktop, repeat the call scenario using CTI OS (if possible) and note if the behavior is different. If it also looks incorrect, proceed to normal UCCE troubleshooting for a CTI issue. If it looks normal proceed to the normal Genesys/T-Server/Desktop troubleshooting. Differences to look for may be things like: incorrect button enablement; incorrect agent state; incorrect call state; mismatched call variables, etc.

Note that as of the time writing of this document that dual desktop for given agents is not supported in production. For testing however using one as ?view? only and the other to initiate actions is acceptable. Although in most cases it?s advisable to just log-out of the desktop after trying a scenario and log-in and try the scenario on the other desktop.

CTI vs Call Setup Issues

When analyzing third party request failures (MAKE_CALL_REQ for example), the error returned may point to UCCE or to Genesys. Look at the type of failure returned. For example, getting an INVALID_CSTA_DEVICE_IDENTIFIER generally indicates an error on the client (Genesys) end. If you get something to the effect of an invalid dialed number, it could be a UCCE configuration. With the preceding CTI OS test, along with trying a similar call scenario on the phone, you can usually ascertain if the problem is with the client or with UCCE.

Note also that if a call fails (CALL_FAILURE_EVENT), it generally indicates that the problem is happening on the UCCE side. Possible causes include configuration or setup issues, or that the passed DN was bad.

Again these can all be found in the CTI Server log. By carefully looking at the error and code, it will help to isolate where the problem is.

Important Information/Special information

This section contains certain information of particular help in pointing out differences for those familiar with troubleshooting CTI Server in Parent/Child deployments and ALL_EVENTS mode.

Trace Mask Level

Since in PERIPHERAL_MONITOR mode events can be filtered and the perspective can be different (by monitor ID) it's generally helpful to have client level messages on. (F8 vs F0) for the EMSTraceMask in CTI-Server. This allows viewing the events as they go to clients and seeing any differences between peripheral monitor IDs.

Useful CTI Server procmon Commands

The following procmon commands are especially useful in diagnosing issues when PERIPHERAL_MONITOR mode is used.

list_monitors (lm)

The lm command will show all the peripheral monitors in effect. You should see listed all agent extensions, all route points (local and remote), and so on. If you don't, there could be some missing or incorrect configuration in the T-Server causing not all devices to be monitored.

Here is an example of the list monitors command:

```
>>>>lm
MonitorID MonitorType Subject Client
1          Device      2501   johno
2          Device      2502   johno
```

One thing worth noting is that from a protocol perspective, Monitor Cross reference IDs are only unique per client. However, CTI Server currently uses unique IDs across all clients, making it a little easier to look at logs.

Genesys_T-Server-UCCE_Integration_Problem_Isolation

dump_device (dd)

Providing similar information as above but on a per device basis is the dump device command. It lists all the monitors on a particular device.

Here is an example of the dump_device command:

```
>>>>dd 2301
```

```
PeripheralID=5000 DeviceType=0(Device) DeviceID=2301 Extension= AgentID=1001 Origin=MonitorStartRe  
MonitorID=3 MonitorType=Device m_Subject:"2301" Client:"(null) "
```

```
>>>>dd 2302
```

```
PeripheralID=5000 DeviceType=0(Device) DeviceID=2302 Extension= AgentID=1002 Origin=SetAgentStateR  
MonitorID=4 MonitorType=Device m_Subject:"2302" Client:"(null) "
```

list_clients (clients)

The clients command shows all the connected clients. You can use the signature to tell if it's a T-Server or not.

```
>>>>clients
```

Session	Time	Ver	Flags	ClientID	AgentID	AgentExt	Signature	Host
1	4 days 14		AUX	CTIOSServer			CTIOSServer	(10.86.140.107:1295)
4	4 days 14		AUX	ACMIPIIM			ACMIPIIM	(10.86.140.106:3152)

list_routing_devices (lrd)

The list_routing_devices command will show you what route points are registered by clients for control and the status. They all should be ROUTE_REGISTER_ACCEPTED ? If not there is an error. In particular you can check if both T-Servers are registered for a route point. Note if this is a CTI-Only deployment with Parent/Child you'll see the ACMI PIM registered for certain route points that the T-Server is not (and cannot be) registered for.

```
>>>>lrd
```

```
PeripheralID ----- DN -----  
----- State ---- SessionIDs  InvokeID  Cause  
5000          2511  
ENABLED             4           5154     0 - ROUTE_REGISTER_ACCEPTED  
5000          1001  
ENABLED             4           5155     0 - ROUTE_REGISTER_ACCEPTED  
5000          1002  
ENABLED             4           5156     0 - ROUTE_REGISTER_ACCEPTED  
5000          1003  
ENABLED             4           5157     0 - ROUTE_REGISTER_ACCEPTED  
5000          1004  
ENABLED             4           5158     0 - ROUTE_REGISTER_ACCEPTED
```

```
5 total routing devices
```

Using CTI-TEST all events mode to compare

If you suspect there is a bug/problem on UCCE with PERIPHERAL_MONITOR mode causing incorrect behavior one way to check is to compare the events that the T-Server gets to those events that an ALL_EVENTS client gets.

If you do this, you can see all events happening on UCCE regardless of peripheral monitors, and so on. If you use CTI-Test to do this it is recommended that you turn the tracing in CTI-TEST OFF for events by doing the following:

```
da /off
ds /off
dc /off
```

Do these prior the issuing an CTI-TEST ?open?. This will turn off tracing of agent/call/session messages. It is not recommended that you do this in a busy system. Once you do this you will have a session log in the CTI-Server log showing all events which you can use for comparison. Note that in a system where there is already an all events client connected (CTI OS, the ACMI PIM, etc.) this is unnecessary.

PERIPHERAL_MONITOR mode

As show in other sections, the T-Server connects using the PERIPHERAL_MONITOR_MODE service, not the ALL_EVENTS service. The biggest difference between PERIPHERAL_MONITOR and ALL_EVENTS is that the T-Server must express interest in devices it wishes to get events for. These are done with MONITOR_START_REQ requests and are confirmed with MONITOR_START_CONF messages. Note that the confirmation contains a MonitorID that is the cross reference seen in events. (MonitorID is 0 for all events when using ALL_EVENTS mode). The T-Server uses this to identify what event the event is ?mostly? for. Also things done involving multiple devices may receive multiple events for the same event (for the same client). Looking at the events is pretty much the same as ALL_EVENTS from a diagnostic point of view. The one difference being if a particular monitor ID gets an event and the local connection state on the event.

Note that PERIPHERAL_MONITOR mode support in CTI Server isn?t new, but has been utilized by only a handful of CTI Server clients over the years.

For example, shown below is a call from agent 1 to agent 2, both have their phones monitored. Look at the highlighted section for the difference from ALL_EVENTS mode. Note that agent events are not shown and not all call events are shown. Note the MonitorID is being populated. Note that AGENT_STATE_EVENTS come up with the MonitorID of the agents phone.

PERIPHERAL_MONITOR mode:

```
16:56:27:612 cg1A-ctisvr SESSION 4: MsgType:BEGIN_CALL_EVENT (MonitorID:2 PeripheralID:5000
PeripheralType:IPCC Simplified
```

```
16:56:27:612 cg1A-ctisvr SESSION 4: NumCTIClients:1 NumNamedVariables:0 NumNamedArrays:0
CallType:CALLTYPE_AGENT_INSIDE
```

```
16:56:27:612 cg1A-ctisvr SESSION 4: ConnectionDeviceIDType:CONNECTION_ID_STATIC
ConnectionCallID:16784707
```

Genesys_T-Server-UCCE_Integration_Problem_Isolation

16:56:27:612 cg1A-ctisvr SESSION 4: CalledPartyDisposition:CD_INVALID ConnectionDeviceID:"2302"
ANI:"2302"

16:56:27:612 cg1A-ctisvr SESSION 4: CTIClientSignature:"johno@JOHNO-WXP01"
CTIClientTimestamp:0x4bcf667b (04/21/10 16:56:27)

16:56:27:612 cg1A-ctisvr SESSION 4: MsgType:CALL_SERVICE_INITIATED_EVENT (MonitorID:2
PeripheralID:5000 PeripheralType:IPCC

16:56:27:612 cg1A-ctisvr SESSION 4: Simplified ConnectionDeviceIDType:CONNECTION_ID_STATIC
ConnectionCallID:16784707

16:56:27:612 cg1A-ctisvr SESSION 4: LineHandle:0 LineType:LINETYPE_INSIDE ServiceNumber:N/A
ServiceID:N/A

16:56:27:612 cg1A-ctisvr SESSION 4: SkillGroupNumber:19201 SkillGroupID:5000 SkillGroupPriority:0

16:56:27:612 cg1A-ctisvr SESSION 4: CallingDeviceType:DEVID_DEVICE_IDENTIFIER
LocalConnectionState:LCS_INITIATE

16:56:27:612 cg1A-ctisvr SESSION 4: EventCause:CEC_NONE ConnectionDeviceID:"2302"
CallingDeviceID:"2302")

16:56:27:658 cg1A-ctisvr SESSION 4: MsgType:CALL_ORIGINATED_EVENT (MonitorID:2
PeripheralID:5000 PeripheralType:IPCC Simplified

16:56:27:658 cg1A-ctisvr SESSION 4: ConnectionDeviceIDType:CONNECTION_ID_STATIC
ConnectionCallID:16784707 LineHandle:0

16:56:27:658 cg1A-ctisvr SESSION 4: LineType:LINETYPE_INSIDE ServiceNumber:N/A ServiceID:N/A
SkillGroupNumber:N/A

16:56:27:658 cg1A-ctisvr SESSION 4: SkillGroupID:N/A SkillGroupPriority:0
CallingDeviceType:DEVID_DEVICE_IDENTIFIER

16:56:27:658 cg1A-ctisvr SESSION 4: CalledDeviceType:DEVID_DEVICE_IDENTIFIER
LocalConnectionState:LCS_INITIATE

16:56:27:658 cg1A-ctisvr SESSION 4: EventCause:CEC_NONE ConnectionDeviceID:"2302"
CallingDeviceID:"2302"

16:56:27:658 cg1A-ctisvr SESSION 4: CalledDeviceID:"2301")

16:56:27:674 cg1A-ctisvr SESSION 4: MsgType:CALL_DELIVERED_EVENT (MonitorID:2
PeripheralID:5000 PeripheralType:IPCC Simplified

16:56:27:674 cg1A-ctisvr SESSION 4: ConnectionDeviceIDType:CONNECTION_ID_STATIC
ConnectionCallID:16784707 LineHandle:N/A

Genesys_T-Server-UCCE_Integration_Problem_Isolation

16:56:27:674 cg1A-ctisvr SESSION 4: LineType:LINETYPE_UNKNOWN ServiceNumber:N/A
ServiceID:N/A SkillGroupNumber:N/A

16:56:27:674 cg1A-ctisvr SESSION 4: SkillGroupID:N/A SkillGroupPriority:0
AlertingDeviceType:DEVID_DEVICE_IDENTIFIER

16:56:27:674 cg1A-ctisvr SESSION 4: CallingDeviceType:DEVID_DEVICE_IDENTIFIER
CalledDeviceType:DEVID_DEVICE_IDENTIFIER

16:56:27:674 cg1A-ctisvr SESSION 4: LastRedirectDeviceType:DEVID_NONE
LocalConnectionState:LCS_INITIATE

16:56:27:674 cg1A-ctisvr SESSION 4: EventCause:CEC_NEW_CALL NumNamedVariables:0
NumNamedArrays:0 ConnectionDeviceID:"2302"

16:56:27:674 cg1A-ctisvr SESSION 4: AlertingDeviceID:"" CallingDeviceID:"2302"
CalledDeviceID:"2301")

16:56:27:674 cg1A-ctisvr SESSION 4: MsgType:BEGIN_CALL_EVENT (MonitorID:1 PeripheralID:5000
PeripheralType:IPCC Simplified

16:56:27:674 cg1A-ctisvr SESSION 4: NumCTIClients:1 NumNamedVariables:0 NumNamedArrays:0
CallType:CALLTYPE_AGENT_INSIDE

16:56:27:690 cg1A-ctisvr SESSION 4: ConnectionDeviceIDType:CONNECTION_ID_STATIC
ConnectionCallID:16784707

16:56:27:690 cg1A-ctisvr SESSION 4: CalledPartyDisposition:CD_INVALID ConnectionDeviceID:"2302"
ANI:"2302" DNIS:"2301"

16:56:27:690 cg1A-ctisvr SESSION 4: DialedNumber:"2301"
CTIClientSignature:"john@JOHNO-WXP01" CTIClientTimestamp:0x4bcf667b

16:56:27:690 cg1A-ctisvr SESSION 4: (04/21/10 16:56:27))

Note 2nd BEGIN_CALL_EVENT sent to Session 4 for MonitorID 1, Previous was for MonitorID 2

16:56:27:690 cg1A-ctisvr SESSION 4: MsgType:CALL_DELIVERED_EVENT (MonitorID:2 PeripheralID:5000 PeripheralType:IPCC Simplified

16:56:27:690 cg1A-ctisvr SESSION 4: ConnectionDeviceIDType:CONNECTION_ID_STATIC
ConnectionCallID:16784707 LineHandle:0

16:56:27:690 cg1A-ctisvr SESSION 4: LineType:LINETYPE_INSIDE ServiceNumber:N/A ServiceID:N/A
SkillGroupNumber:N/A

16:56:27:690 cg1A-ctisvr SESSION 4: SkillGroupID:N/A SkillGroupPriority:0
AlertingDeviceType:DEVID_DEVICE_IDENTIFIER

16:56:27:690 cg1A-ctisvr SESSION 4: CallingDeviceType:DEVID_DEVICE_IDENTIFIER
CalledDeviceType:DEVID_DEVICE_IDENTIFIER

PERIPHERAL_MONITOR mode

Genesys_T-Server-UCCE_Integration_Problem_Isolation

16:56:27:690 cg1A-ctisvr SESSION 4: LastRedirectDeviceType:DEVID_NONE
LocalConnectionState:LCS_CONNECT EventCause:CEC_NONE

16:56:27:690 cg1A-ctisvr SESSION 4: NumNamedVariables:0 NumNamedArrays:0
ConnectionDeviceID:"2302" AlertingDeviceID:"2301"

16:56:27:690 cg1A-ctisvr SESSION 4: CallingDeviceID:"2302" CalledDeviceID:"2301")

16:56:27:690 cg1A-ctisvr SESSION 4: MsgType:CALL_DELIVERED_EVENT (MonitorID:1
PeripheralID:5000 PeripheralType:IPCC Simplified

16:56:27:690 cg1A-ctisvr SESSION 4: ConnectionDeviceIDType:CONNECTION_ID_STATIC
ConnectionCallID:16784707 LineHandle:0

16:56:27:690 cg1A-ctisvr SESSION 4: LineType:LINETYPE_INSIDE ServiceNumber:N/A ServiceID:N/A
SkillGroupNumber:N/A

16:56:27:690 cg1A-ctisvr SESSION 4: SkillGroupID:N/A SkillGroupPriority:0
AlertingDeviceType:DEVID_DEVICE_IDENTIFIER

16:56:27:690 cg1A-ctisvr SESSION 4: CallingDeviceType:DEVID_DEVICE_IDENTIFIER
CalledDeviceType:DEVID_DEVICE_IDENTIFIER

16:56:27:690 cg1A-ctisvr SESSION 4: LastRedirectDeviceType:DEVID_NONE
LocalConnectionState:LCS_ALERTING EventCause:CEC_NONE

16:56:27:690 cg1A-ctisvr SESSION 4: NumNamedVariables:0 NumNamedArrays:0
ConnectionDeviceID:"2302" AlertingDeviceID:"2301"

16:56:27:690 cg1A-ctisvr SESSION 4: CallingDeviceID:"2302" CalledDeviceID:"2301")

Note 2 CALL_DELIVERED_EVENTS being sent. One for each monitored party in the call (Source and Dest). Also note that the source is LCS_CONNECT, and the destination is LCS_ALERTING (CSTA like).

16:56:30:877 cg1A-ctisvr SESSION 4: MsgType:CALL_ESTABLISHED_EVENT (MonitorID:2 PeripheralID:5000 PeripheralType:IPCC Simplified

16:56:30:877 cg1A-ctisvr SESSION 4: ConnectionDeviceIDType:CONNECTION_ID_STATIC
ConnectionCallID:16784707 LineHandle:0

16:56:30:877 cg1A-ctisvr SESSION 4: LineType:LINETYPE_INSIDE ServiceNumber:N/A ServiceID:N/A
SkillGroupNumber:19201

16:56:30:877 cg1A-ctisvr SESSION 4: SkillGroupID:5000 SkillGroupPriority:0
AnsweringDeviceType:DEVID_DEVICE_IDENTIFIER

16:56:30:877 cg1A-ctisvr SESSION 4: CallingDeviceType:DEVID_DEVICE_IDENTIFIER
CalledDeviceType:DEVID_DEVICE_IDENTIFIER

16:56:30:877 cg1A-ctisvr SESSION 4: LastRedirectDeviceType:DEVID_NONE
LocalConnectionState:LCS_CONNECT EventCause:CEC_NONE

Genesys_T-Server-UCCE_Integration_Problem_Isolation

16:56:30:877 cg1A-ctisvr SESSION 4: ConnectionDeviceID:"2301" AnsweringDeviceID:"2301"
CallingDeviceID:"2302"

16:56:30:877 cg1A-ctisvr SESSION 4: CalledDeviceID:"2301")

16:56:30:877 cg1A-ctisvr SESSION 4: MsgType:CALL_ESTABLISHED_EVENT (MonitorID:1
PeripheralID:5000 PeripheralType:IPCC Simplified

16:56:30:877 cg1A-ctisvr SESSION 4: ConnectionDeviceIDType:CONNECTION_ID_STATIC
ConnectionCallID:16784707 LineHandle:0

16:56:30:893 cg1A-ctisvr SESSION 4: LineType:LINETYPE_INSIDE ServiceNumber:N/A ServiceID:N/A
SkillGroupNumber:19201

16:56:30:893 cg1A-ctisvr SESSION 4: SkillGroupID:5000 SkillGroupPriority:0
AnsweringDeviceType:DEVID_DEVICE_IDENTIFIER

16:56:30:893 cg1A-ctisvr SESSION 4: CallingDeviceType:DEVID_DEVICE_IDENTIFIER
CalledDeviceType:DEVID_DEVICE_IDENTIFIER

16:56:30:893 cg1A-ctisvr SESSION 4: LastRedirectDeviceType:DEVID_NONE
LocalConnectionState:LCS_CONNECT EventCause:CEC_NONE

16:56:30:893 cg1A-ctisvr SESSION 4: ConnectionDeviceID:"2301" AnsweringDeviceID:"2301"
CallingDeviceID:"2302"

16:56:30:893 cg1A-ctisvr SESSION 4: CalledDeviceID:"2301")

16:56:30:893 cg1A-ctisvr Trace:

Note 2 CALL_ESTABLISHED_EVENTS being sent. One for each monitored party in the call (Source and
Dest). Note here that the connection state of each is LCS_CONNECT.

16:56:36:346 cg1A-ctisvr SESSION 4: MsgType:CALL_CONNECTION_CLEARED_EVENT (MonitorID:2 PeripheralID:5000 PeripheralType:IPCC

16:56:36:346 cg1A-ctisvr SESSION 4: Simplified ConnectionDeviceIDType:CONNECTION_ID_STATIC
ConnectionCallID:16784707

16:56:36:346 cg1A-ctisvr SESSION 4: ReleasingDeviceType:DEVID_DEVICE_IDENTIFIER
LocalConnectionState:LCS_NULL

16:56:36:346 cg1A-ctisvr SESSION 4: EventCause:CEC_NONE ConnectionDeviceID:"2302"
ReleasingDeviceID:"2302")

16:56:36:346 cg1A-ctisvr SESSION 4: MsgType:CALL_CONNECTION_CLEARED_EVENT
(MonitorID:1 PeripheralID:5000 PeripheralType:IPCC

16:56:36:346 cg1A-ctisvr SESSION 4: Simplified ConnectionDeviceIDType:CONNECTION_ID_STATIC
ConnectionCallID:16784707

Genesys_T-Server-UCCE_Integration_Problem_Isolation

16:56:36:346 cg1A-ctisvr SESSION 4: ReleasingDeviceType:DEVID_DEVICE_IDENTIFIER
LocalConnectionState:LCS_CONNECT

16:56:36:346 cg1A-ctisvr SESSION 4: EventCause:CEC_NONE ConnectionDeviceID:"2302"
ReleasingDeviceID:"2302")

16:56:36:362 cg1A-ctisvr SESSION 4: End callID 16784707.2302(s) at periph 5000 device 2302

Note above that we get 2 CALL_CONNECTION_CLEARED events for the calling party dropping ? 1 for each monitored device in the call.

Note below that we only get 1 (only one monitored party left in the call). Also notice that two END_CALL_EVENTS are sent, one for each monitored party in the call.

16:56:36:362 cg1A-ctisvr SESSION 4: MsgType:END_CALL_EVENT (MonitorID:2 PeripheralID:5000 PeripheralType:IPCC Simplified

16:56:36:362 cg1A-ctisvr SESSION 4: ConnectionDeviceIDType:CONNECTION_ID_STATIC
ConnectionCallID:16784707

16:56:36:362 cg1A-ctisvr SESSION 4: ConnectionDeviceID:"2302")

16:56:36:362 cg1A-ctisvr SESSION 4: MsgType:CALL_CONNECTION_CLEARED_EVENT
(MonitorID:1 PeripheralID:5000 PeripheralType:IPCC

16:56:36:362 cg1A-ctisvr SESSION 4: Simplified ConnectionDeviceIDType:CONNECTION_ID_STATIC
ConnectionCallID:16784707

16:56:36:362 cg1A-ctisvr SESSION 4: ReleasingDeviceType:DEVID_DEVICE_IDENTIFIER
LocalConnectionState:LCS_NULL

16:56:36:362 cg1A-ctisvr SESSION 4: EventCause:CEC_NONE ConnectionDeviceID:"2301"
ReleasingDeviceID:"2301")

16:56:36:393 cg1A-ctisvr SESSION 4: MsgType:END_CALL_EVENT (MonitorID:1 PeripheralID:5000
PeripheralType:IPCC Simplified

16:56:36:393 cg1A-ctisvr SESSION 4: ConnectionDeviceIDType:CONNECTION_ID_STATIC
ConnectionCallID:16784707 16:56:36:393 cg1A-ctisvr SESSION 4: ConnectionDeviceID:"2301")

Other differences in PERIPHERAL_MONITOR mode have to do with when a BEGIN_CALL_EVENT and other events are seen/sent to the client. For a normal ACD call processed by UCCE you won't see the call begin until either it gets queued or hits an agent. The CALL_QUEUED_EVENT will come up on the monitor ID of the last Route Point the call hit.