

# 思科的MediaSense常见问题

## 目录

### [简介](#)

### [先决条件](#)

### [要求](#)

### [使用的组件](#)

- [1. 如何关联不同的呼叫情形的参考呼叫ID，在统一通信管理器电话分叉下？](#)
  - [1.1. MediaSense搜索和作用呼叫关联功能](#)
  - [1.2. 代理程序保留/恢复方案](#)
  - [1.3. 客户保留/恢复方案](#)
  - [1.4. 代理程序转移到另一个代理程序方案](#)
  - [1.5. 与另一个代理程序方案的代理程序会议](#)
- [2. 如何关联不同的呼叫会话的参考呼叫ID，在Unified边界网元分叉下？](#)
  - [2.1. MID呼叫编码崔凡吉莱](#)
  - [2.2. 咨询传输](#)
  - [2.3. 参见呼叫检测](#)
  - [2.4. 参见从广泛参加者的呼叫检测](#)
  - [2.5. 摘要](#)
- [3. 如何连结呼叫在思科MediaSense中与他们的在其他解决方案组件的外观？](#)
  - [3.1. 标识符相关表](#)
- [4. 如何确定哪跟踪有主叫方，并且哪跟踪有被叫方？](#)
  - [4.1. 对于呼叫由多维数据集分叉了](#)
  - [4.2. 对于呼叫由Unified CM电话分叉了](#)
- [5. 什么是CLOSED\\_ERROR的会话状态的可能的原因？](#)
- [6. 被修剪的和删除的会话有何区别？](#)
  - [6.1. 使用getAllPrunedSessions查询](#)
  - [6.2. 使用getSessions查询](#)
  - [6.3. 为什么行为差异在被修剪的和删除的会话上？](#)
- [7. 如何配置梅迪亚分叉的一个TDM网关？](#)
- [8. 如何捕获实际目的地电话，当曾经搜索组时？](#)
- [9. 统一通信管理器基于网络录音为什么是推荐作为首选的分叉机制？](#)
- [10. 节点更加长为什么采取升级到MediaSense 10.5？](#)
- [11. 什么是俄国时间区域更改的影响在MediaSense搜索和作用应用程序？](#)
- [12. 什么是MediaSense支持的语言？](#)
- [13. 如何监控MediaSense系统性能？](#)
- [14. 如何配置浏览器运行浏览器播放机在MediaSense？](#)

## 简介

本文描述Cisco媒介感觉服务器的常见问题。

## [先决条件](#)

## 要求

Cisco 建议您了解以下主题：

- 思科MediaSense
- Cisco Unified Communications管理器(CUCM)

## 使用的组件

本文档中的信息基于以下软件和硬件版本：

思科MediaSense 10.5

# 1.如何关联不同的呼叫情形的参考呼叫ID，在统一通信管理器电话分叉下？

在思科MediaSense中，每呼叫的元数据只提供xRefCi (参考呼叫ID)和设备ref (分机)分叉的设备和远端设备(可以是会议桥或任何其他电话)。

xRefCi参数是一特定的媒体流的统一的通信管理器的标识符。他们总是不对应1:1与已录制跟踪。

## 1.1. MediaSense搜索和作用呼叫关联功能

MediaSense生成在暂挂/恢复或者转移的情况下被记录，使困难识别呼叫的所有录音会话的呼叫的多个会话。为了能关联单个呼叫的这些录音会话，MediaSense介绍作为呼叫关联被叫做的新特性。通过此功能，与一个普通的xRefci值的所有严格相关的呼叫一起分组。MediaSense 10.5支持构件在网桥录音的呼叫关联功能。

## 1.2.代理程序保留/恢复方案

1. 代理程序A (extn 1000)和呼叫方C (extn 2000)在通话状态互相呼叫和。
2. 代理程序A呼叫暂挂。
3. 代理程序A恢复呼叫。

有此方案的两记录的会话：

- 有sessionId = S1的会话与这两跟踪，为了在代理程序前的时间/segment呼叫暂挂：  
trackNumber = 0与参加者A (deviceRef = 1000， xRefCi = aaaa)  
trackNumber = 1与参加者B (deviceRef = 2000年， xRefCi = cccc)
- 有sessionId = S2的会话与这两跟踪，为了时间/分段，在代理程序恢复呼叫后。  
trackNumber = 0与参加者A (deviceRef = 1000， xRefCi = aaaa)  
trackNumber = 1与参加者B (deviceRef = 2000年， xRefCi = cccc)

当代理程序呼叫暂挂时，MediaSense不记录呼叫的分段。

## 1.3.客户保留/恢复方案

1. 代理程序A (extn 1000)和呼叫方C (extn 2000)在通话状态alled和。
2. 客户C呼叫暂挂。

3. 客户C恢复呼叫。

整个呼叫在一会话上被记录此方案的：

- 有sessionId = S1的会话与这两跟踪：

- trackNumber = 0与参加者A (deviceRef = 1000 , xRefCi = aaaa)

- trackNumber = 1与参加者B (deviceRef = 2000年 , xRefCi = cccc)

在此方案中，当客户呼叫暂挂时，MediaSense也记录呼叫的分段。

## 1.4.代理程序转移到另一个代理程序方案

1. 呼叫方C (extn 2000)呼叫代理程序A (extn 1000)

2. 代理程序A (extn 1000)用代理程序B (3000)咨询

3. 代理程序A (extn 1000)完成转移。

4. 代理程序B (extn 3000)挂断。

使用统一通信管理器9.x和前，这是结果：

1. Caller C (extn 2000)呼叫代理程序A (extn 1000)

会话S1开始，跟踪0是A (extn 1000)，跟踪1是C (extn 2000)。

对另一个代理程序B (extn 3000)的2. Agent A (extn 1000)转接呼叫。A和B设备为分叉设置。

3. Caller C (extn 2000)听到Music on Hold (MoH)。

4. Agent A (extn 1000)与B (extn 3000)谈。

- 会话S1被结束

- 开始的会话S2，跟踪0是A (extn 1000)，跟踪1是B (extn 3000)

- 会话S3开始，跟踪0是B (extn 3000)，跟踪1是A (extn 1000)

考虑事项：

- 会话S1末端由于代理程序A电话暂挂呼叫方C。

- 因为两个电话为分叉，配置塞申斯S2和S3存在。

- 参加者和xRefCi两个参加者的在S2和S3是相同的，但是在反向位置从彼此。

- 因为咨询认为一独立呼叫，xRefCi值S1在S2或S3没有反射。

5. Agent A (extn 1000)完成转移。

6. C (extn 2000)谈与B (extn 3000)。

7. A (extn 1000)断开了。

结束的会话S2。

会话S3更新和跟踪0是B (extn 3000)，并且跟踪1是C (extn 2000)。

考虑事项：

- 一次远端的转移触发现有录制会话的更新。

- 远端的参加者更改对那S1。

- S3新建的远端的xRefCi匹配S1远端的xRefCi。

- 代理程序B (extn 3000)挂断。
- C (extn 2000)和B (extn 3000)被断开。
- 会话S3被结束

**注意：**一次远端的转移导致现有的会话的更新。分叉的电话在跟踪0在跟踪1更改依然是唯一的参加者，但是参加者对新的当事人。

**在统一通信管理器10.0的情况下及以后，这是结果：**

1. 呼叫方C (extn 2000)呼叫代理程序A (extn 1000)。

C (extn 2000)与代理程序A (extn 1000)谈。会话S1 -开始-跟踪0是A (extn 1000)，跟踪1是C (2000)

2. 代理程序A (extn 1000)参见代理程序B (extn 3000)。

3. C (extn 2000)听到MoH。A (extn 1000)与B (extn 3000)谈。

- 会话S1 -结束
- 会话S2 -开始-跟踪0是A (extn 1000)，跟踪1是B (extn 3000)
- 会话S3 -开始-跟踪0是B (extn 3000)，跟踪1是A (extn 1000)

**考虑事项：**

- 因为代理程序A电话暂挂呼叫方C，会话S1结束。
- 因为两个电话为分叉，配置塞申斯S2和S3两个存在。
- 参加者以及xRefCi两个参加者的在S2和S3相同，但是在被倒转的位置从彼此。
- 因为咨询认为一独立呼叫，S1 xRefCi值在S2或S3没有反射

4. 代理程序A (extn 1000)完成转移。

5. C (extn 2000)与B (extn 3000)谈。

6.A (extn 1000)断开了。

- 会话S3 -结束
- 会话S2 -结束
- 会话S4 -开始-跟踪0是B (extn 3000)，跟踪1是C (extn 2000)

**考虑事项：**

- 一次远端的转移触发一录制会话结尾和另一录制会话开始。
- 虽然个新会话开始，其xRefCi值将匹配上次会话。
- S4远端的xRefCi匹配S1远端的xRefCi，并且S4近端的xRefCi匹配S3近端的xRefCi。

7. 代理程序B (extn 3000)挂断。

8. C (extn 2000)和B (extn 3000)断开了。

- 会话S4 -结束

**注意：**一次远端的转移导致一录制会话结尾和另一录制会话开始。

## 1.5. 与另一个代理程序方案的代理程序会议

1. 呼叫方C (extn 2000)呼叫代理程序A (extn 1000)。
2. 代理程序A (extn 1000)参见代理程序B (extn 3000)。
3. 代理程序A (extn 1000)完成会议。
4. 代理程序A (extn 1000)从会议丢弃。
5. 代理程序B (extn 3000)挂断。

**在统一通信管理器9.x的情况下和前，这是结果：**

1. 呼叫方C (extn 2000)呼叫代理程序A (extn 1000)。
2. C (extn 2000)与A (extn 1000)谈。  
会话S1开始-跟踪0是A (extn 1000)，跟踪1是C (extn 2000)。
3. 代理程序A (extn 1000)参见代理程序B (extn 3000)。
4. C (extn 2000)听到MoH A (extn 1000)谈话对B (extn 3000)

- 会话S1被结束
- 开始的会话S2 -跟踪0是A (extn 1000)，跟踪1是B (extn 3000)
- 会话S3开始-跟踪0是B (extn 3000)，跟踪1是A (extn 1052)

**考虑事项：**

- 因为代理程序A电话暂挂呼叫方C，会话S1结束。
  - 因为两个电话为分叉，配置塞申斯S2和S3存在。
  - 参加者以及xRefCi两个参加者的在S2和S3相同，但是在被倒转的位置从彼此。
  - 因为咨询认为一独立呼叫，S1 xRefCi值在S2或S3没有反射。
5. 代理程序A (extn 1000)完成会议。
  6. C (extn 2000)谈与A (extn 1000)和B (extn 3000)。

- 结束的会话S2
- 会话S3更新-跟踪0是B (ext 3000)，跟踪1是会议桥
- 会话S4开始-跟踪0是A (extn 1000)，跟踪1是会议桥

**考虑事项：**

一次远端的转移触发现有录制会话的更新。

会议的完成实现：

在咨询期间：

- 咨询的电话暂挂一主要的呼叫和激活参见呼叫。
- 参见的电话只有一个激活的呼叫(咨询呼叫)。

当会议完成(连接的所有当事人)：

- 咨询的电话的参见呼叫终止。
- 咨询的电话的主要的呼叫获得一次远端的转移到会议桥。
- 参见的电话获得一次远端的转移到会议桥。

**结果：**

- S2终止，因为表示咨询的电话的参见呼叫，也终止。
- S4开始;它代表A的主要的呼叫继续和远端的转移，但是原始S1不可能更新，因为以前是终止的由于及早保持。

- S3更新的获得，因为B的远端从A转接到会议桥。
- S4近端的xRefCi值将匹配S1近端的xRefCi值。

7. 代理程序A (extn 1000)从会议丢弃。

8. A (extn 1000)断开了。 C (extn 2000)谈与B (extn 3000)会话S3更新-跟踪0是B (extn 3000)，跟踪1是C (extn 2000)。

**考虑事项：**

- 一个会议的降级到一正常两大政党呼叫里实现作为彼此转接两个剩余的电话的远端
- 一次远端的转移触发现有录制会话的更新。
- S3和S1将有匹配近端的xRefCi值。注意仅一会话依然是活动，因为呼叫方C没有启用的分叉。

9. 代理程序B (extn 3000)挂断。

10. C (extn 2000)和B (extn 3000)断开了。  
会话S4被结束。

**考虑事项：**

- 一次远端的转移导致现有的会话的更新。分叉的电话在跟踪0在跟踪1更改依然是唯一的参加者，但是参加者对新的当事人。
- 会议创建与所有电话转移到会议桥。所以，会议操作正如一套转移。现有的会话在分叉的电话依然是唯一的参加者在跟踪0的那些会话，但是参加者在跟踪1更改更新对会议桥。
- 一旦第三方从会议下降，当事人彼此转接。这再更新现有的会话，分叉的电话在跟踪0在跟踪1更改依然是唯一的参加者，但是参加者对另一个当事人。
- 如果第四个当事人被添加到会议桥，没有在元数据的征兆，除非第四个当事人也有其启用的自己分叉。

**在统一通信管理器10.x的情况下和以后，这是结果：**

1. 呼叫方C (extn 2000)呼叫代理程序A (extn 1000)。

2. C (extn 2000)谈与A (extn 1000)会话S1 -开始-跟踪0是A (extn 1000)，跟踪1是C (extn 2000)。

3. 代理程序A (extn 1000)参见代理程序B (extn 3000)。

4. C (extn 2000)听到MoH A (extn 1000)谈与B (extn 3000)。

- 会话S1 -结束
- 会话S2 -开始-跟踪0是A (extn 1000)，跟踪1是B (extn 3000)
- 会话S3 -开始-跟踪0是B (extn 3000)，跟踪1是A (extn 1000)

**考虑事项：**

- 因为代理程序A的电话暂挂呼叫方C，会话S1结束。
- 因为两个电话为分叉，配置塞申斯S2和S3存在。
- 参加者以及xRefCi两个参加者的在S2和S3相同，但是在被倒转的位置从彼此。
- 因为咨询认为一独立呼叫，S1 xRefCi值在S2或S3没有反射

5. 代理程序A (extn 1000)完成会议。

6. C (extn 2000)谈与A (extn 1000)和B (extn 3000)

- 会话S2 -结束
- 会话S3 -结束
- 会话S4 -开始-跟踪0是A (extn 1000) , 跟踪1是会议桥
- 会话S5 -开始-跟踪0是B (extn 3000) , 跟踪1是会议桥

#### 考虑事项：

一次远端的转移触发一录制会话结尾和另一录音开始。会议的完成实现列出了得此处：

- 在咨询期间：咨询的电话暂挂—主要的呼叫和激活参见呼叫参见的电话只有一个激活的呼叫(咨询呼叫)
- 当会议完成(连接的所有当事人)：
  - 咨询的电话的参见呼叫终止
  - 咨询的电话的主要的呼叫获得一次远端的转移到会议桥
  - 参见的电话获得一次远端的转移到会议桥
- 结果：因为代理程序A和代理程序B有启用的分叉两新的会话创建S4近端的xRefCi值和S1近端的xRefCi值匹配S5近端的xRefCi值和S3近端的xRefCi值匹配S4和S5的远端xRefCi值不配比，即使两个连接到同一座会议桥

7. 代理程序A (extn 1000)从会议丢弃

8. A (extn 1000)断开了。 C (extn 2000)谈与B (extn 3000)

- 会话S4 -结束
- 会话S5 -结束
- 会话S6 -开始-跟踪0是B (extn 3000) , 跟踪1是C (extn 2000)

#### 考虑事项：

- 一个会议的降级到—正常两大政党呼叫里实现作为彼此转接两个剩余的电话的远端
- 一次远端的转移触发一录制会话结尾和另一录制会话开始
- S6和S5将有匹配近端的xRefCi值。注意仅一会话依然是活动，因为呼叫方C没有启用的分叉

9. 代理程序B (extn 3000)挂断

10. C (extn 2000)和B (extn 3000)断开了

- 会话S6 -结束

#### 考虑事项：

- 一次远端的转移导致一会话末端和别的开始
- 会议创建与所有电话转移到会议桥。所以，会议操作正如一套转移。结束现有的会话，并且新建的会话创建在分叉的电话和会议桥之间
- 一旦第三方从会议下降，当事人彼此转接。这结束包括会议桥并且启动在两个剩余的终端之间的新的会话的会话
- 如果第四个当事人被添加到会议桥，没有在元数据的征兆，除非第四个当事人也有其启用的自己分叉

## 2.如何关联不同的呼叫会话的参考呼叫ID，在Unified边界网元分叉下？

当Unified边界网元分叉，非常少量情况导致一呼叫拆分到广泛录音会话。保留/恢复、传送和会议操

作不开始新建记录会话在大多数情况下。在新的会话创建的少量案件中，有共用值，CCID (呼叫相关性ID)。此值对呼叫的所有会话是普通。CCID是思科GUID十进制表，由思科语音路由器生成的一唯一呼叫密钥。收到呼叫的第一个路由器生成此密钥，并且通过它在线路下到所有随后的设备包括思科MediaSense。

Unified边界网元不创造xRefCi值，但是创建与统一通信管理器电话分叉的呼叫的相似性，思科MediaSense也综合一个对每Unified边界网元呼叫的xRefCi值。这些在跟踪级别的元数据能被看到，与CCID一起，出现在会话级别。

这些情况原因统一将拆分的边界网元录音到多个会话：

## 2.1.MID呼叫编码崔凡吉莱

如果转移、会议、会议丢弃，或者其他操作造成当事人重新协商他们的编码，思科MediaSense结束当前录音会话并且开始新的。两会话共享同样CCID和同一个对xRefCi值。

## 2.2. 咨询传输

咨询传输是从一个代理程序的一次转移到另一个，两个代理程序彼此谈，当原始呼叫方暂挂中时等待。呼叫的咨询段与整体呼叫在某个方面涉及，并且配置统一通信管理器这样是可能的请参见呼叫穿过多维数据集。然而，Unified边界网元和思科MediaSense不知道这些呼叫涉及，并且他们创建新的CCID和一个新的对此会话的xRefCi值。

这些呼叫可以互相关联与比较参加者deviceRef和时间戳字段。考虑此方案：

1. 呼叫方C (extn 2000)呼叫代理程序A (extn 1000) (sessionId = S1, CCID = C1)
2. 代理程序A用代理程序B (extn 3000) (sessionId = S2, CCID = C2)咨询
3. 代理程序A丢包和呼叫方C与代理程序B (sessionId = S1、CCID = C1)谈

在此方案的红旗在步骤2。在该期限，代理程序A (deviceRef 1000)立即是一个参加者在两记录的会话：

- 会话= S1/CCID = C1和
- 会话= S2/CCID = C2

所以，S1与S2涉及，并且C1与C2涉及。

## 2.3. 参见呼叫检测

首先，我们需要清楚的定义参见呼叫：

由一个当前参加者在现有的会话做终端的是外部会话，并且在该会话排除其他参加者的任何附属呼叫。

在理论上，此方案可能包括代理程序暂挂呼叫方用他的上司检查luch工间休息时间，甚至代理程序呼叫方暂挂收到从他的妻子的一呼叫，但是我们暂时忽略那些可能性。

是可能的为了客户端应用能由思科MediaSense事件数据流的跟踪检测一咨询呼叫在实时。如果客户端观察会话开始的事件包含一给的deviceRef，与另一个会话开始的事件包含同一deviceRef没有干预的对话端事件，能认为，在两个会话开始的事件和CCIDs找到的sessionIds关联。

历史上，客户端能检查其中任一参见关联与一给的主要的呼叫的呼叫，与思科MediaSense API。假



设客户端认识在CCID <C1>的该代理程序A使用的extn 1000。此说明查找任何已关联参见呼叫：

Step1. 通过发出getSessionByCCID(<C1>)获取主要的呼叫的会话元数据。

Step2. 解压缩sessionStartDate (呼叫它<Ta>)和sessionDuration。

Step3. 通过添加sessionDuration计算sessionEndDate (呼叫它<Tb>)到<Ta>。

Step4. 运行此API请求：

```
https://Mediasense IP
address:8443/ora/queryService/query/getSessionsByDeviceRef?value=1000&minSessionStartDate=<Ta>&maxSessionStartDate=<Tb>
```

此查询能返回超过一会话。如果它，则所有可以假设关联与同一呼叫。

## 2.4. 参见从广泛参加者的呼叫检测

被提及的步骤参见呼叫检测部分，查找全部参见由接收最初的电话的设备做的呼叫。然而，那里若参见由呼叫是随后转接的设备做的呼叫？

考虑此步骤：

1. 呼叫方呼叫Agent1
2. Agent1用代理程序2咨询，然后丢包
3. 呼叫方用代理程序2发言
4. 代理程序2用代理程序3咨询，然后丢包
5. 呼叫方用代理程序3发言

此步骤不捉住在代理程序2和代理程序3之间的咨询呼叫。

因为这是一Unified边界网元呼叫，我们能利用事实所有呼叫方和其中每一个的连接代理程序之间在同样录制会话包括和事实介入的所有代理程序在同一会话列出作为参加者曾经。因此，从主要的会话元数据，我们能收集是包含的所有deviceRefs的列表。要找到那些会话，我们能做一系列的呼叫到getSessionByDeviceRef，与每请求一deviceRef一起指定主要的会话的时间范围。

或者，进程可以简化与单个getSession请求例如此：

```
{
  "requestParameters": [
    {
      "fieldName": "deviceRef",
      "fieldConditions": [
        {
          "fieldOperator": "equals",
          "fieldValues": [
            "1000"
          ],
          "fieldConnector": "OR"
        },
        {
          "fieldOperator": "equals",
          "fieldValues": [
            "2000"
          ],
          "fieldConnector": "OR"
        }
      ]
    }
  ]
}
```

```

    },
    {
        "fieldOperator": "equals",
        "fieldValues": [
            "3000"
        ],
        "fieldConnector": "OR"
    },
    {
        "fieldOperator": "equals",
        "fieldValues": [
            "4000"
        ]
    }
],
"paramConnector": "AND"
},
{
    "fieldName": "sessionStartDate",
    "fieldConditions": [
        {
            "fieldOperator": "between",
            "fieldValues": [
                <Ta>, // session start time
                <Tb> // session end time
            ]
        }
    ]
}
]
}
]
}

```

此查询回所有咨询电话关联与原始主要的呼叫和所有其转移。

此步骤太宽广地实际上转换网。如果例如，在deviceRef 4000的代理程序执行并且终止偶然开始，在<Ta>后的一完全独立的呼叫，并且，在他被添加了到有问题的前的呼叫，此步骤包括独立请呼叫在集。此问题可以虽则解决，与在主要的会话的元数据的有用的资料。每个参加者的信息包括他加入会话和他的使用权的持续时间的的时间偏移。客户端代码可能使用删除的信息从接收上述的列表的无关的会话。或者，它可能公式化正确地成帧时间在期间每个代理程序在主要的呼叫的一系列的直接getSessions或getSessionByDeviceRef查询。我们离开作为练习到读者。

## 2.5.摘要

在前面的部分，我们提交了所有会话检索的precedures关联与一给的思科MediaSense录制会话。然而，我们也看到一给的呼叫可能分开成超过一会话，和一旦MID呼叫编码更改。

如何获取所有录音(以及咨询)关联与所有会话连接对呼叫方的交互作用？

答案是延伸此多个detectiion的说明参见呼叫。首先，我们收集共享有问题的主要的会话CCID的所有会话。然后，我们创建我们的参与者名单从所有的那些会话记录。其次，我们计算时间范围作为最早期的会话的sessionStartDate通过最新的会话的末端。最后，我们能执行显示的getSessions查询。

和前面，我们能最终获得许多录音捕获，因此我们可能完成后加工步骤删除从其列表的那些无关的会话。

## 3.如何连结呼叫在思科MediaSense中与他们的在其他解决方案组

# 件的外观？

## 3.1. 标识符相关表

这两个表——一个统一通信管理器呼叫的和一个Unified边界网元呼叫的。每列代表解决方案组件或协议，与第一列代表思科MediaSense。每行代表标识符特定类型。

要读表，请开始与代表数据项您知道，然后查找对列水平地代表解决方案组件您要查找呼叫的信元。在该信元的条目由什么名称指示确切同样数据项在目标组件被认识。如果目标组件有一个空白的信元在该行，则该数据项没有为该组件所知。您能寻找您能交叉垂直到另一行该信元不是空白的在目标组件的列的干预的列。

例如，与统一通信管理器呼叫，假设，您认识GED-188 CallReferenceID，并且您在Cisco MediaSense中要查找呼叫。检查左侧从GED-188列，您看到没有在MediaSense列的值，因此您不能映射直接地到它。

然而，有您能在行间曲折前行的列：统一通信管理器CDR。客户端能通过搜索选择适当的统一通信管理器CDR记录IncomingProtocolCallRef匹配GED-188 CallReferenceID的一个。记录包含值在思科MediaSense中呼叫destLegCallIdentifier，因此是相同的象MediaSense NearEnd xRefCi，并且可能使用查找对应的记录。

统一通信管理器CDR记录没有写入，直到一些时间，在呼叫的结尾终止，然而，因此后可能历史上只使用此方法。

有另一个路径。向下从GED-188 CallReferenceID查找。它结果您能也使用AlertingDevice和AnsweringDevice匹配deviceRef字段在MediaSense。此方法在实时也运作。

Call Correlation for Calls Forked by a Unified CM IP Phone

MediaSense	Ingress Gateway or CUBE	AAA RADIUS CDR	UCM CDR	TAPI/JTAPI field	UCCE Database	UCCE Script	CTIOS	GED-188
(1)	Cisco-GUID		IncomingProtocolCallRef	CiscoConnection.UniqueID	TCD.CallGUID	Call.CallGUID		CallReferenceID
NearEnd xRefCi			destLegCallIdentifier	Terminal.ConnectionID				
FarEnd xRefCi			origLegCallIdentifier	Terminal.ConnectionID				
			global_CallID_call-ManagerId + global_CallID_callId (A.K.A. UCM GCID)	CiscoCall.CallID	TCD.PeripheralCallKey			
deviceId							Agent.AgentInstrument	
deviceRef					TCD.InstrumentPortNumber (2)		Agent.AgentExtension or Agent.Extension	AlertingDevice or AnsweringDevice

Call Correlation for Calls Forked by CUBE

MediaSense	Ingress Gateway or CUBE	AAA RADIUS CDR	UCM CDR	TAPI/JTAPI field	UCCE Database	UCCE Script	CTIOS	GED-188
CCID (3)	Cisco-GUID	Cisco-GUID	IncomingProtocolCallRef		TCD.CallGUID	Call.CallGUID		CallReferenceID
deviceRef	Called or calling party extn				TCD.InstrumentPortNumber (2)		Agent.AgentExtension or Agent.Extension	AlertingDevice or AnsweringDevice

### 考虑事项：

1. 在统一通信管理器呼叫录音，思科MediaSense实际上接收从UCM的一思科GUID，但是不是由其他解决方案设备捕获的同样一个。因此MediaSense没有均等存储此值。

2. 对于代理对代理呼叫，TCD.InstrumentPortNumber是目的地代理扩展。呼叫的代理扩展可以在TCD.ANI找到。
3. CCID是思科GUID以十进制形式，是4连字符被分离的套10位十进制数字。这些可以转换到有每10位十进制数字的转换的六角形的表对一个八位数字的六角形的编号的，并且删除连字符。那里思科GUID用于UCCE，以其六角形的形式。

## 4.如何确定哪跟踪有主叫方，并且哪跟踪有被叫方？

### 4.1.对于呼叫由多维数据集分叉了

对于多维数据集呼叫，请跟踪0总是映射对锚点段媒体流。锚点段是记录配置文件的媒体配置的dial-peer。第二个跟踪地图非锚点段。

如果有记录配置文件的媒体启用在入站dialpeer，则锚点段变为段。换句话说，主叫方在跟踪0出现，并且被叫方在跟踪1.出现。

如果有记录配置文件的媒体启用在出站dialpeer，则锚点段变为外段。在那种情况下主叫方在跟踪1出现，并且被叫方在跟踪0出现。

### 4.2.对于呼叫由Unified CM电话分叉了

对于分叉，在简单呼叫情形的Unified CM您在元数据能使用xRefCi字段确定哪个当事人在哪媒体请跟踪。数字上更加小的xRefCi通常是指主叫方的跟踪。被叫方的跟踪数字上更加大(通常由一个，但是它可能更多在一个合理加载的系统下)。然而，这些xRefCi值最终包裹到零。因此，如果发现一个值是大量，并且其他是一个小编号，您假设他们的位置被倒转。

在更加复杂的方案中，此算法总是不运作。如果附加服务被调用，例如转移和会议，并且UC管理器集群包括超过一个节点，则xRefCi值顺序地不一定生成，并且您不能假设，他们的顺序有所有含义。直接方式确定一个特定的对的排序顺序xRefCi值是否可以委托将查看xRefCi值的第一个字节。此字节代表该特定的标识符创建的UC管理器节点ID。如果两个xRefCi值的第一个字节是相同的，则他们的排序正确。如果他们不同的，则排序也许不正确。

对于这些案件，确定呼叫方向的唯一方法在实时是获取从其他来源的信息，例如JTAPI事件源。一旦呼叫结束，并且几分钟流逝了，您能总是确定呼叫方向和检查UC管理器的CDR数据为呼叫。特别地，CDR记录的origLegcallIdentifier字段总是代表呼叫方。

## 5.什么是CLOSED\_ERROR的会话状态的可能的原因？

CLOSED\_ERROR的会话状态的可能的原因包括：

1. 呼叫控制服务器接收从媒体(录音)服务器的一个错误反应开放的或Close请求。
2. 呼叫控制服务器检测SIP信号错误，例如丢失ACK。
3. 会话顺利地关闭，但是所有跟踪有零的大小。

当会话在活动状态时，是正常没有在元数据的持续时间，因为持续时间不知道，直到会话关闭。

对于在CLOSED\_ERROR状态的会话，如果会话或跟踪持续时间字段不是存在事件或getSessions数据，然后此跟踪的媒体不是可用的。

## 6. 被修剪的和删除的会话有何区别？

考虑这两查询：

### 6.1. 使用getAllPrunedSessions查询

此查询返回一套会话，会话状态删除的所有：

```
https://Mediaserver IP address:8443/ora/queryService/query/getAllPrunedSessions?minSessionStartDate=1301788800000&maxSessionStartDate=1312329599000
```

### 6.2. 使用getSessions查询

此查询不返回会话：

```
https://MediaServer IP address:8443/ora/queryService/query/getSessions
```

```
{
  "requestParameters":
  [{
    "fieldName" : "sessionState",
    "fieldConditions":
    [{
      "fieldOperator" : "equals",
      "fieldValues" : [ "DELETED" ]
    }],
    "paramConnector" : "AND"
  },
  {
    "fieldName" : "sessionStartDate",
    "fieldConditions":
    [{
      "fieldOperator" : "between",
      "fieldValues" : [ "1301788800000", "1312329599000" ]
    }]
  }
]
```

### 6.3. 为什么行为差异在被修剪的和删除的会话上？

行为差异是故意的。参考此在MediaSense文档的部分：

- API参数描述：getAllPrunedSessions API说明：

请使用此API搜索所有被修剪的录音...被修剪的期限是指录音由思科MediaSense系统删除。使用deleteSessions API，如果明确地删除任何记录，则这些删除的录音没有考虑作为被修剪的录音。

- 在部分积极的存储管理下的MediaSense SRND：

当会话被修剪时，关联与这些会话的元数据在数据库保持，在这些会话被标记作为‘修剪’以后。此元数据不采取很多存储空间与录音比较，但是采取若干空间，并且应该周期地删除。要帮助在此活动，客户端可能周期地发行一个API要求被修剪的会话，或者客户端可能决定接收会话被修剪的事件和明确地删除客户端不再需要的那些事件。

要澄清，两查询是完全不同的。实际上，第二查询(lwhich搜索状态删除)的所有会话总是返回空集。正常每日查询过滤会话以删除的状态，即使那是什么是请求的。唯一的例外是 **getAllPrunedSessions**。此例外是打算的协助应用程序找到被修剪的会话，以便应用程序能请求这些会话删除。

一旦使用在您从**getAllPrunedSessions**获得被修剪的会话的列表的**deleteSessions** API，这些会话在**getAllPrunedSessions**结果不再出现。这样会话完全从元数据立即删除。

另一个方式查看此是被修剪的会话不是事和删除的会话一样：

1. 被修剪的会话为删除被指示了由在MediaSense系统的一种算法。人在决策未涉及修剪这些会话。因此，即使这些会话被迁移向删除的状态，这些会话没有从元数据实际上删除。人(或应用程序)干预要求。由于这些会话在删除的状态，这些会话不是可视对多数查询。然而，这些会话是可视对**getAllPrunedSessions**查询API。并且，如果任何mp4文件为这些会话生成，这些mp4文件继续是存在磁盘和继续可以下载，直到被修剪的会话实际上删除。
2. 删除的会话通过明确地呼叫**deleteSessions** API指示。此标记可以对做对已经被修剪的会话或未删除的会话。一旦会话由**deleteSessions** API删除，此会话不再是可视对所有查询。这包括**getAllPrunedSessions** API。这些删除的会话从元数据立即删除，以便磁盘空间可以恢复。

## 7.如何配置梅迪亚分叉的一个TDM网关？

当您有呼叫流和您要记录那些呼叫的一个PSTN网关。这些呼叫是TDM对SIP呼叫。然而，媒体分叉只是可用的在SIP对SIP呼叫。

这些呼叫可以被记录。这些呼叫能处理通过路由器每第二次。配置指导和其他详细信息可以在[此白皮书](#)找到。

## 8.如何捕获实际目的地电话，当曾经搜索组时？

当您使用分叉从多维数据集时的媒体，MediaSense元数据通常包含被叫方的分机。然而，如果数字称为是通信管理器搜索组实验号码，默认情况下然后元数据只包含该实验号码。它不包含实际上应答呼叫电话的分机。

有能更改此的通信管理器设置。在搜索/试验配置页，请查找部分题为**连接的Party转换**。必须打开设置显示行组成员DN作为已连接Party。

此功能可用在通信管理器9.0(1)及以后。

## 9. 统一通信管理器基于网络录音为什么是推荐作为首选的分叉机制？

使用统一通信管理器基于网络录音(NBR)，您能使用网关记录呼叫。不管设备、位置或者地理，NBR允许统一通信管理器路由录音呼叫。使用NBR，呼叫记录媒体可以来源从IP电话或从连接给在

SIP中继的统一通信管理器的网关。管理器动态地选择正确的媒体的统一通信来源基于呼叫流和呼叫参加者。

NBR提供一自动fallback构件在网桥(围嘴)，当集成服务路由器(ISR)时不可用，因为分开的记录的配置没有要求。这是有用的，在客户要包括代理程序代理程序参见在录音策略处的呼叫，因为Unified边界网元不能记录参见呼叫，因此围嘴需要分开启用。

使用xRefci，NBR和围嘴呼叫可以关联，从统一通信管理器JTAPI是可得到。CISCO-GUID不是需要的，含义CTI服务器和CTIOS连接没有要求。尽管有单个相关性标识符，在组件间的相关性更加强，并且可以执行用一个统一方式对立于呼叫流。

使用NBR，直接拨号的以及拨号程序启动的呼出能关联与他们的在其他解决方案组件的外观。

使用NBR，TDM网关录音自动地使用没有路由器的产能的已分解。目前，TDM网关录音不支持与MediaSense 10.5。

## 10.节点更加长为什么采取升级到MediaSense 10.5？

节点能耗费几个小时升级取决于拿着的数量和大小录音。对于MediaSense 10.5，当您升级与非常大数据集时的一个节点，需要大约每1百万录音的90另外的分钟。

## 11.什么是俄国时间区域更改的影响在MediaSense搜索和作用应用程序？

MediaSense的用户搜索，并且作用应用程序受影响，如果他们在查找任何被影响的时间区域或，如果他们选择搜索条件的一被影响的时间区域。协调与MediaSense的第三方合作伙伴产品类似受影响，直到他们更新他们的各自时间区域表。

应急方案是选择匹配从GMT的正确偏移量的时间区域，即使城市不再正确。

## 12.什么是MediaSense支持的语言？

这是MediaSense支持的语言：

- 阿拉伯
- 丹麦
- 荷兰
- 英语(美国)
- 芬兰
- 法国
- 德国
- 意大利
- 日本
- 大韩
- 挪威
- 波兰
- 巴西葡萄牙语
- 俄罗斯

- 简体中文
- 西班牙
- 瑞典
- 繁体中文
- 土耳其

## 13. 如何监控MediaSense系统性能？

要监控MediaSense系统性能，请分析值这些关键性能指示器(KPIs)用RTMT工具或思科最初协作保证工具。

关于RTMT工具或思科头等协作保证工具的更多信息，请参考[思科MediaSense用户指南的Unified RTMT管理和思科头等协作保证管理部分](#)。

关键绩效指标(KPI)	KPIs和他们的阈值	
	RTMT计数器	建议的阈值
呼叫成功率	MediaSense呼叫控制Service>记录数会话不出错误	99.99%
	MediaSense呼叫控制Service>记录数有错误的会话	呼叫成功率=记录数会话不出 * 100
API API的答复平均时间	思科MediaSense API Service>平均值查询响应时间	60秒
记录起始平均值设置延迟	思科MediaSense呼叫控制Service>平均值设置延迟	3秒
CPU平均值利用率	处理器> % CPU时间	90%
内存平均利用率	使用的内存> %Mem	70%
RTP/UDP丢包	丢弃的网络接口> Rx > eth0	0
	网络接口> Rx错误> eth0	0

## 14. 如何配置浏览器运行浏览器播放机在MediaSense？

凭浏览器，请执行这些步骤运行浏览器播放机：

### Internet Explorer 9

1. 在MediaSense搜索和作用，请点击play图标录制会话。

2. 点击是委托证书。

注意：验证提供的自签名证书是被瞄准的MediaSense节点通过验证在证书的技术详细资料的FQDN。

### Internet Explorer 11

前提：在新安装MediaSense的情况下11.0，请保证使用各自完全合格的域名(FQDN)，MediaSense节点被添加到集群。

在对MediaSense 11.0的升级的情况下，请保证使用主机名，以前被添加应该由各自FQDN当前显示的MediaSense节点。检查在MediaSense服务器配置窗口(思科MediaSense管理>System的MediaSense服务器列表>MediaSense服务器配置)。

请执行下列步骤：

1. 设置集合hostnameformediaurl CLI作为“真”做MediaSense准备mediaurls的mp4url和其余使用仅FQDN。

```
admin:set useHostNameForMediaURL
```

### Mozilla Firefox

1. 添加一个MediaSense节点的名称证书端口的mp4url 8446在Mozilla Firefox的委托权限的。

2. 添加自签名证书，点击录制的下载图标并且选择mp4。此连接是不信任弹出窗口出现



```
admin:set useHostNameForMediaURL true
```

2. 重新启动配置服务激活属性。

```
admin:utils service restart Cisco  
MediaSense Configuration Service
```

**注意：**如果服务未适当地重新启动，再请执行同一命令。

3. 在配置服务重新启动后，请签字并且签到对**MediaSense搜索并且使用**。

**限制：**使用IP，万一MediaSense节点以前被添加了，然后节点继续由IP仅显示在升级以后到MediaSense 11.0。浏览器播放机在Internet Explorer 11不运转，不考虑*hostnameformediaurl* CLI value命令的。在这种情况下，推荐不应该设置CLI命令的

*hostnameformediaurl*作为“真”。

执行对Windows的MediaSense自签名证书委托权限的以下步骤添加。

1. 打开**MediaSense搜索并且使用**。

安全证书弹出窗口出现。

2. 单击**继续**。

**MediaSense搜索和play**窗口出现。

3. 在地址栏，请点击**验证错误**图标。

4. 单击**视图证书**。

证书弹出窗口出现。

5. 单击**安装证书**。

证书导入向导出现。

6. 单击**其次**。

7. 在**证书存储**窗口，请选择地方在以下**存储**单选按钮的**所有证书**并且单击**浏览**。

**挑选证书存储**对话框出现。

8. 检查**Show physical**存储复选框并且选择**可信的根Certificationhorities**文件夹。

9. 单击**OK**键和**其次**。

10. 单击**芬通社**完成证书导入。

安全警告弹出窗口看上去确认证书的安装。

11. 单击**是**。

下列信息出现。

```
The import was successful.
```

12. 单击**Ok**。

13. 单击**OK**键在证书弹出窗口的。

14. 单击**Close**和打开浏览器。

15. 单击**MediaSense搜索和作用**。

安全证书issxvi仍然仍然存在。

16. 单击**工具> Internet选项**

**>Advanced**，不选定关于**证书地址不匹配**检查boxviiunder安全的**警告**。

17. 单击**适用**并且**好**。

18. 重新启动浏览器和打开

3. 单击**play**图标与选定录音相应。浏览器播放机播放选定录制会话。

2. 添加自签名证书，单击录制的**下载**图标并且选择**mp4**。**此连接是不信任**弹出窗口出现。**注意：**验证提供的自签名证书是**准的MediaSense节点通过验证证书的技术详细资料的FQDN**。

## MediaSense搜索并且使用。

保证MediaSense服务器通过在浏览器的FQDN是可取得。否则，请导航对C:\Windows\System32\drivers\etc，打开在Notepad的HOSTS文件并且在文件的底部添加MediaSense服务器和其FQDN的IP地址。浏览器播放机开始工作在Internet Explorer 11。

**注意：**如果录音是存在集群的一个不同的MediaSense节点，您是被提示的添加该MediaSense节点证书在委托权限的。

3. 点击**我了解风险链路**。
4. 单击**添加例外**。  
添加安全例外弹出窗口出现。
5. 点击**确认安全例外**。  
8446端口特定MS节点的自签名证书  
添加到浏览器的委托权限。