

# Cisco MediaSense のための FAQ

## 目次

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

1. [どのように Unified Communications Manager 電話分岐の下で異なるコール シナリオのための参照呼び出しID を、関連させますか。](#)
  - 1.1. [MediaSense 検索および演劇コール アソシエーション 機能](#)
  - 1.2. [エージェント保持/レジューム シナリオ](#)
  - 1.3. [顧客保持/レジューム シナリオ](#)
  - 1.4. [別のエージェント シナリオへのエージェント転送](#)
  - 1.5. [別のエージェント シナリオのエージェント会議](#)
2. [どのように Unified Border Element 分岐の下で異なるコール セッションのための参照呼び出しID を、関連させますか。](#)
  - 2.1. [中間コール コーデック変更](#)
  - 2.2. [コンサルト転送](#)
  - 2.3. [コール 検出を参照して下さい](#)
  - 2.4. [複数の参加要素からのコール 検出を参照して下さい](#)
  - 2.5. [要約](#)
3. [どのように他のソリューション コンポーネントの外観と Cisco MediaSense のコールを関連付けますか。](#)
  - 3.1. [識別相関表](#)
4. [どのトラックがコーリングパーティがあり、どのトラックが被呼加入者があるかどのように判別しますか。](#)
  - 4.1. [コールに関しては CUBE によって分岐しました](#)
  - 4.2. [コールに関しては統一された CM 電話によって分岐しました](#)
5. [CLOSED ERROR のセッション状態のための原因とは何か。](#)
6. [プルーニングされ、削除されたセッション間の違いとは何か。](#)
  - 6.1. [getAllPrunedSessions クエリを使って](#)
  - 6.2. [getSessions クエリを使って](#)
  - 6.3. [なぜプルーニングされ、削除されたセッションの動作違いか。](#)
7. [メディア分岐のための TDM ゲートウェイの設定方法か。](#)
8. [ハントする グループを使用する場合の実際の宛先電話をキャプチャする方法か。](#)
9. [メカニズムはなぜ Unified Communications Manager ネットワークベース記録好まれた分岐として推奨されてありますか。](#)
10. [ノードはなぜ MediaSense 10.5 にアップグレードするために時間がかかりますか。](#)
11. [MediaSense 検索および演劇アプリケーションのロシア時間帯変更の影響とは何か。](#)
12. [MediaSense によってサポートされる言語とは何か。](#)
13. [MediaSense システムパフォーマンスを監視する方法か。](#)
14. [ブラウザを MediaSense の内部ブラウザ プレイヤーを実行するために設定する方法か。](#)

## 概要

この資料は Ciscoメディア感覚サーバのための FAQ を記述したものです。

## 前提条件

### 要件

次の項目に関する知識が推奨されます。

- Cisco MediaSense
- Cisco Unified Communication Manager ( CUCM )

### 使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づくものです。

Cisco MediaSense 10.5

## 1 の L-AC-PLS-G ) を注文します。どのように Unified Communications Manager 電話分岐の下で異なるコール シナリオのための参照呼び出しID を、関連させますか。

Cisco MediaSense では、各コールのためのメタデータは xRefCi ( 参照呼び出しID ) および分岐デバイスおよび遠端デバイスのデバイス ref ( 拡張だけ ) を提供します ( コンファレンスブリッジまたは他のどの電話もある場合もあります )。

xRefCi パラメータは特定のメディア ストリームのための統合された通信マネージャの識別です。それらは記録されたトラックと常に 1:1 対応しません。

### 1.1. MediaSense 検索および演劇コール アソシエーション 機能

MediaSense はコールのすべての記録セッションを識別すること困難にする、保持/レジュームまたは転送の場合には記録されるコールのための複数のセッションを生成します。これらに関連付けられます単一の呼び出しの記録セッションは、MediaSense コール アソシエーションとして命名される新しい機能を導入します。この機能を通して、よくある xRefci 値のすべての強く関連するコールは一緒にグループ化されます。MediaSense 10.5 は構築ブリッジ 記録物のためのコール アソシエーション 機能をサポートします。

### 1.2. エージェント保持/レジューム シナリオ

1. エージェント A ( 1000 ) extn および発信者 C ( extn は話す状態で 2000 ) 互いをおよび呼び出しました。
2. エージェント A はコールを保留にします。
3. エージェント A 再開コール。

このシナリオのための 2 記録セッションがあります:

- sessionId = これら二つのトラックが付いている S1 とのセッション、なぜならエージェントの前の時間 /segment はコールを保留にします:

trackNumber = 参加要素 A ( deviceRef = 1000、 xRefCi = aaaa ) との 0  
trackNumber = 参加要素 B ( deviceRef = 2000 年、 xRefCi = cccc ) との 1

- sessionId = これら二つのトラックとの S2 のセッション、なぜならエージェントの後の時間 / セグメントはコールを再開します。

trackNumber = 参加要素 A ( deviceRef = 1000、 xRefCi = aaaa ) との 0  
trackNumber = 参加要素 B ( deviceRef = 2000 年、 xRefCi = cccc ) との 1

MediaSense はエージェントがコールを保留にする間、コールのセグメントを記録しません。

### 1.3. 顧客保持/レジューム シナリオ

1. エージェント A ( 1000 ) extn および発信者 C ( extn は話す状態で 2000 ) 互いをおよび alled。
2. 顧客 C はコールを保留にします。
3. 顧客 C 再開コール。

このシナリオのための全体のコールは 1 セッションに記録されます:

- sessionId = これら二つのトラックが付いている S1 とのセッション:  
trackNumber = 参加要素 A ( deviceRef = 1000、 xRefCi = aaaa ) との 0  
trackNumber = 参加要素 B ( deviceRef = 2000 年、 xRefCi = cccc ) との 1

このシナリオで MediaSense はまた顧客がコールを保留にする間、コールのセグメントを記録します。

### 1.4. 別のエージェント シナリオへのエージェント転送

1. 発信者 C ( extn は 2000 ) エージェント A ( 1000 ) extn を呼出します
2. エージェント A ( extn はエージェント B によって 1000 ) 相談します ( 3000 )
3. エージェント A ( extn は 1000 ) 転送を完了します。
4. エージェント B ( extn は 3000 ) ハングアップします。

Unified Communications Manager 9.x で先に、結果はここにあり、:

1. Caller C ( extn は 2000 ) エージェント A ( 1000 ) extn を呼出します

開始するセッション S1 はトラック 0 A です ( 1000 ) extn は、トラック 1 C ( 2000 ) extn です。

2. Agent A ( extn 1000 ) 転送は別のエージェント B ( 3000 ) extn に呼出します。 A および B 両方デバイスは分岐のために設定されます。

3. Caller C ( extn は 2000 ) Music on Hold ( MoH を聞きます ) 。

4. Agent A ( extn は B ( 3000 ) extn に 1000 ) 話します。

- 終了するセッション S1
- 開始するセッション S2 はトラック 0 A です ( 1000 ) extn は、トラック 1 B ( 3000 ) extn です
- 開始するセッション S3 はトラック 0 B です ( 3000 ) extn は、トラック 1 A ( 1000 ) extn です

**考慮事項:**

- エージェント A 電話によるセッション S1 端は発信者 C を保留にしました。
- 電話が両方とも分岐のために設定されるのでセッション S2 および S3 はあります。
- S2 および S3 の 2 人の参加要素のための参加要素および xRefCi は互いからの逆位置で同一、しかしです。
- S1 の xRefCi 値は S2 か S3 に相談が独立したコールとみなされるので、反映されません。

5.Agent A ( extn は 1000 ) 転送を完了します。

6.C ( 2000 ) B ( 3000 ) extn に話す extn。

7.A ( 1000 ) 切られる extn。

終了するセッション S2。

アップデートされるセッション S3 およびトラック 0 は B です ( extn 3000 ) およびトラック 1 は C ( 2000 ) extn です。

#### 考慮事項:

- 遠端転送は既存の記録セッションのアップデートを誘発します。
- 遠端参加要素は S1 のそれに変更します。
- S3 新しい遠端 xRefCi は S1 遠端 xRefCi と一致します。
- エージェント B ( extn は 3000 ) ハングアップします。
- C ( extn 2000 ) および B ( extn は 3000 ) 切断されています。
- 終了するセッション S3

注: 遠端転送は既存のセッションのアップデートという結果に終わります。分岐電話はトラック 0 に唯一の参加要素、新しいパーティへのトラック 1 変更の参加要素に残ります。

#### Unified Communications Manager 10.0 および以降の場合には、結果はここにあります:

1. 発信者 C ( extn は 2000 ) エージェント A ( 1000 ) extn を呼出します。

C ( extn はエージェント A ( 1000 ) extn に 2000 ) 話します。セッション S1 -開始される-トラック 0 は A です ( 1000 ) extn は、トラック 1 C です ( 2000 )

2. エージェント A ( extn によっては 1000 ) エージェント B ( 3000 ) extn が参照します。

3. C ( extn は 2000 ) MoH を聞きます。A ( extn は B ( 3000 ) extn に 1000 ) 話します。

- セッション S1 -終了される
- セッション S2 -開始される-トラック 0 は A です ( 1000 ) extn は、トラック 1 B ( 3000 ) extn です
- セッション S3 -開始される-トラック 0 は B です ( 3000 ) extn は、トラック 1 A ( 1000 ) extn です

#### 考慮事項:

- エージェント A 電話が発信者 C を保留にしたのでセッション S1 は終了します。
- セッションは S2 および S3 両方電話が両方とも分岐のために設定されるのであります。
- S2 および S3 の 2 人の参加要素のための参加要素、また xRefCi は同一しかし互いからの反転させた位置にです。

• S1 xRefCi 値は S2 か S3 に相談が独立したコールとみなされるので、反映されません  
4. エージェント A ( extn は 1000 ) 転送を完了します。

5. C ( extn は B ( 3000 ) extn に 2000 ) 話します。

6. A ( 1000 ) 切られる extn。

- セッション S3 -終了される
- セッション S2 -終了される
- セッション S4 -開始される-トラック 0 は B です ( 3000 ) extn は、トラック 1 C ( 2000 ) extn です

#### 考慮事項:

- 遠端転送は 1 記録セッションの終わりおよび別の記録セッションの開始するを誘発します。
- 新しいセッションが開始するけれども、xRefCi 値は前のセッションを一致する。
- S4 遠端 xRefCi は S1 遠端 xRefCi と一致し、S4 近端 xRefCi は S3 近端 xRefCi と一致します。

7. エージェント B ( extn は 3000 ) ハングアップします。

8. C ( extn 2000 ) および B ( 3000 ) 切られる extn。

- セッション S4 -終了される

注: 遠端転送は 1 記録セッションの終わりおよび別の記録セッションの開始するという結果に終わります。

## 1.5. 別のエージェント シナリオのエージェント会議

1. 発信者 C ( extn は 2000 ) エージェント A ( 1000 ) extn を呼出します。
2. エージェント A ( extn によっては 1000 ) エージェント B ( 3000 ) extn が参照します。
3. エージェント A ( extn は 1000 ) 会議を完了します。
4. エージェント A ( extn は会議から 1000 ) 廃棄します。
5. エージェント B ( extn は 3000 ) ハングアップします。

Unified Communications Manager 9.x の場合には先に、結果はここにあり、:

1. 発信者 C ( extn は 2000 ) エージェント A ( 1000 ) extn を呼出します。

2. C ( extn は A ( 1000 ) extn に 2000 ) 話します。

開始するセッション S1 -トラック 0 は A です ( 1000 ) extn は、トラック 1 C ( 2000 ) extn です。

3. エージェント A ( extn によっては 1000 ) エージェント B ( 3000 ) extn が参照します。

4. C ( extn は 2000 ) MoH A を聞きます ( extn は B ( 3000 ) extn に 1000 ) 話します

- 終了するセッション S1
- 開始するセッション S2 -トラック 0 は A です ( 1000 ) extn は、トラック 1 B ( 3000 ) extn です
- 開始するセッション S3 -トラック 0 は B です ( 3000 ) extn は、トラック 1 A ( extn 1052 ) です

#### 考慮事項:

- エージェント A 電話が発信者 C を保留にしたのでセッション S1 は終了します。
  - 電話が両方とも分岐のために設定されるのでセッション S2 および S3 はあります。
  - S2 および S3 の 2 人の参加要素のための参加要素、また xRefCi は同一しかし互いからの反転させた位置にです。
  - S1 xRefCi 値は S2 か S3 に相談が独立した コールとみなされるので、反映されません。
5. エージェント A ( extn は 1000 ) 会議を完了します。
6. C ( 2000 ) A ( extn 1000 ) およびに B ( 3000 ) extn 話す extn.

- 終了するセッション S2
- アップデートされるセッション S3 -トラック 0 は B です ( 3000 ) ext は、トラック 1 コンファレンスブリッジです
- 開始するセッション S4 -トラック 0 は A です ( 1000 ) extn は、トラック 1 コンファレンスブリッジです

#### 考慮事項:

遠端転送は既存の記録セッションのアップデートを誘発します。

会議の完了は設定されています:

相談の間:

- 相談電話はプライマリ コールおよびアクティブなコンサルト コールを保留にします。
- 参照した電話に 1 つのアクティブ コール ( コンサルト コール ) がありますただ。

会議が完了する時 ( 接続されるすべてのパーティ ):

- 相談電話のコンサルト コールは終わります。
- 相談電話のプライマリ コールはコンファレンスブリッジへの遠端転送を得ます。
- 参照した電話はコンファレンスブリッジへの遠端転送を得ます。

その結果:

- S2 はまた終わる相談電話のコンサルト コールを表すので、終わります。
  - S4 は開始します; それは A のプライマリ コールの継続および遠端転送を表しますが、先に保持すること以前に終わられた当然があったのでオリジナル S1 は更新済である場合もありません。
  - 遠端 B's が A からコンファレンスブリッジに単に転送されているのでアップデートされる S3 gets。
  - S4 近端 xRefCi 値は S1 近端 xRefCi 値を一致する。
7. エージェント A ( extn は会議から 1000 ) 廃棄します。

8. A ( 1000 ) 切られる extn。 C ( 2000 ) B ( アップデートされる 3000 ) extn セッション S3 話す extn -にトラック 0 は B です ( 3000 ) extn は、トラック 1 C ( 2000 ) extn です。

#### 考慮事項:

- 正常な両者コールへの会議の段階的縮小は互いに両方の残りの電話の遠端転送として設定されます
- 遠端転送は既存の記録セッションのアップデートを誘発します。
- S3 におよび S1 に一致する近端 xRefCi 値があります。 発信者 C に有効になる分岐がないので 1 セッションだけアクティブのままになることに注目して下さい。

9. エージェント B ( extn は 3000 ) ハングアップします。

10. C ( extn 2000 ) および B ( 3000 ) 切られる extn。  
終了するセッション S4。

**考慮事項:**

- 遠端転送は既存のセッションのアップデートという結果に終わります。分岐電話はトラック 0 に唯一の参加要素、新しいパーティへのトラック 1 変更の参加要素に残ります。
- 会議はコンファレンスブリッジへのすべての電話の転送で作成されます。従って、会議は一組の転送と同様に機能します。既存のセッションは分岐電話がトラック 0 に唯一の参加要素に残るそれらのセッション、コンファレンスブリッジへのトラック 1 変更の参加要素でアップデートされます。
- サードパーティが会議から廃棄すれば、パーティは互いに転送されます。これは既存のセッションを再度アップデートします、分岐電話はトラック 0 に唯一の参加要素、他へのトラック 1 変更の参加要素にパーティ残ります。
- 第 4 パーティがコンファレンスブリッジに追加される場合、第 4 パーティがまた自身の分岐を有効にしてもらわなければ、メタデータに示す値がありません。

**Unified Communications Manager 10.x および以降の場合には、結果はここにあります:**

1. 発信者 C ( extn は 2000 ) エージェント A ( 1000 ) extn を呼出します。

2. C ( 2000 ) 話す extn A に ( 1000 ) extn セッション S1 は-開始する-トラック 0 A です ( 1000 ) extn は、トラック 1 C ( 2000 ) extn です。

3. エージェント A ( extn によっては 1000 ) エージェント B ( 3000 ) extn が参照します。

4. C ( 2000 ) MoH A ( 1000 ) B ( 3000 ) extn に話す extn を聞く extn。

- セッション S1 -終了される
- セッション S2 -開始される-トラック 0 は A です ( 1000 ) extn は、トラック 1 B ( 3000 ) extn です
- セッション S3 -開始される-トラック 0 は B です ( 3000 ) extn は、トラック 1 A ( 1000 ) extn です

**考慮事項:**

- エージェント A 電話が発信者 C を保留にしたのでセッション S1 は終了します。
  - 電話が両方とも分岐のために設定されるのでセッション S2 および S3 はあります。
  - S2 および S3 の 2 人の参加要素のための参加要素、また xRefCi は同一しかし互いからの反転させた位置にです。
  - S1 xRefCi 値は S2 か S3 に相談が独立した コールとみなされるので、反映されません
5. エージェント A ( extn は 1000 ) 会議を完了します。

6. C ( 2000 ) A ( extn 1000 ) およびに B ( 3000 ) extn 話す extn

- セッション S2 -終了される
- セッション S3 -終了される
- セッション S4 -開始される-トラック 0 は A です ( 1000 ) extn は、トラック 1 コンファレンスブリッジです

- セッション S5 -開始される-トラック 0 は B です ( 3000 ) extn は、トラック 1 コンファレンスブリッジです

**考慮事項:**

遠端転送は 1 記録セッションの終わりおよび別の記録の開始するを誘発します。 会議の完了はここにリストされている設定されたあります:

- 相談の間: 相談電話はプライマリ コールおよびアクティブなコンサルト コールを保留にします参照した電話にありますただ 1 つのアクティブ コール ( コンサルト コール ) が
- 会議が完了する時 ( 接続されるすべてのパーティ ) :
  - 相談電話のコンサルト コールは終わります
  - 相談電話のプライマリ コールはコンファレンスブリッジへの遠端転送を得ます
  - 参照した電話はコンファレンスブリッジへの遠端転送を得ます
- その結果: 2 新しいセッションはエージェントに A およびエージェント両方 B に有効になる分岐があるので作成されます S4 近端 xRefCi 値および S1 近端 xRefCi は一致を評価します S5 近端 xRefCi 値および S3 近端 xRefCi は一致を評価します S4 および S5 の遠端 xRefCi 値は両方とも同じコンファレンスブリッジに接続されるのに、一致する

7. エージェント A ( extn は会議から 1000 ) 廃棄します

8. A ( 1000 ) 切られる extn。 C ( 2000 ) B ( 3000 ) extn に話す extn

- セッション S4 -終了される
- セッション S5 -終了される
- セッション S6 -開始される-トラック 0 は B です ( 3000 ) extn は、トラック 1 C ( 2000 ) extn です

**考慮事項:**

- 正常な両者コールへの会議の段階的縮小は互いに両方の残りの電話の遠端転送として設定されます
- 遠端転送は 1 記録セッションの終わりおよび別の記録セッションの開始するを誘発します
- S6 におよび S5 に一致する近端 xRefCi 値があります。 発信者 C に有効になる分岐がないので 1 セッションだけアクティブのままになることに注目して下さい

9. エージェント B ( extn は 3000 ) ハングアップします

10. C ( extn 2000 ) および B ( 3000 ) 切られる extn

- セッション S6 -終了される

**考慮事項:**

- 遠端転送は 1 セッションの終わりおよび別のものの開始するという結果に終わります
- 会議はコンファレンスブリッジへのすべての電話の転送で作成されます。 従って、会議は一組の転送と同様に機能します。 既存のセッションは終了し、新しいセッションは分岐電話とコンファレンスブリッジの間で作成されます
- サードパーティが会議から廃棄すれば、パーティは互いに転送されます。 これはコンファレンスブリッジが含まれ、2 つの残りのエンドポイント間の新しいセッションを開始するセッションを終了します



- 第 4 パーティがコンファレンスブリッジに追加される場合、第 4 パーティがまた自身の分岐を有効にしてもらわなければ、メタデータに示す値がありません

## 2. どのように Unified Border Element 分岐の下で異なるコールセッションのための参照呼び出し ID を、関連させますか。

分岐していて Unified Border Element が非常に少数の状況はコールを多重記録セッションに分割します。、転送保持して下さい/レジュームおよび会議オペレーションは新しい記録セッションをほとんどの場合開始しません。新しいセッションが作成されたり少数のケース共通の価値観が、CCID (呼び出し相関 ID) あるか。この値はコールのすべてのセッションによくあります。CCID は Cisco 音声 ルータによって生成される CiscoGUID の Decimal 形式、ユニークな Call 鍵です。コールを受信する最初のルータはこのキーを生成し、行の下で Cisco MediaSense を含むすべてのそれに続くデバイスに渡します。

Unified Border Element 自体は xRefCi 値を生成しませんが、類似性を Unified Communications Manager 電話分岐コールで作成するために、Cisco MediaSense はまた各 Unified Border Element コールの xRefCi 値のペアを総合します。これらはセッション レベルで現われる CCID と共にトラックレベルのメタデータで、見られる場合があります。

これらの状況は Unified Border Element 記録物を複数のセッションに分割します:

### 2.1. 中間コール コーデック変更

転送により、会議、会議ドロップする、または他のオペレーションがパーティがコーデックを再取り決めすれば Cisco MediaSense は現在の記録セッションを終了し、新しいものを開始します。2 つのセッション 共有 xRefCi 値の同じ CCID および同じペア。

### 2.2. コンサルト転送

コンサルト転送はオリジナル発信者が保留の状態で作っている間、1 つのエージェントからの 2 つのエージェントが互いに話す別のものへの転送です。コールの相談レグは全面的なコールと何らかのかたちで関連し、コールをするパススルー CUBE を参照しなさいこと Unified Communications Manager を設定することは可能性のあるそのような物です。ただしこれらのコールが関係している、このセッションの xRefCi 値の新しい CCID および新しいペアを作成しますことを、Unified Border Element および Cisco MediaSense は知らないし。

これらのコールは参加要素 deviceRef の比較し、時刻を記録しますフィールドをと互いに関連付けることができます。次のシナリオについて考えます。

1. 発信者 C ( extn は 2000 ) エージェント A ( 1000 ) extn を呼出します ( sessionId = S1、CCID = C1 )
2. エージェント A はエージェント B ( 3000 ) extn によって相談します ( sessionId = S2、CCID = C2 )
3. エージェント A ドロップおよび発信者 C はエージェント B ( sessionId = S1、CCID = C1 ) によって話します

このシナリオの赤旗はステップ 2. にあります。その期間の間、エージェント A ( deviceRef は 1000 ) 2 記録セッションのすぐに参加要素です:

- セッション = S1/CCID = C1

- セッション = S2/CCID = C2

従って、S1 は C2 と S2 および C1 と関連していません関連していません。

## 2.3. コール 検出を参照して下さい

最初に、コンサルト コールのクリア定義を必要とします:

セッションそのセッションの他の参加要素を除き、外部であるエンドポイントへの既存のセッションの現在の参加要素によってなされるセカンダリ コール。

理論では、このシナリオは luch 中断を彼の主任によってチェックするために保留にします発信者をエージェントを含む可能性がありますまた更にエージェントは彼の妻からコールを受信するために発信者を保留にしましたがそれらの可能性を今のところ無視します。

それは Cisco MediaSense イベント ストリームのトラックによってリアルタイムのコンサルト コールを検出するクライアントアプリケーションのために可能性のあるです。クライアントがセッションを観察すれば開始されたイベントは介入セッションによって終了されるイベント無しで別のセッションによって開始されるイベントのある特定の deviceRef が、含まれています同じ deviceRef が含まれています、2つのセッションによって開始されるイベントで見つけられる sessionIds および CCIDs が準であることを結論できます。

従来、クライアントによってはあるように Cisco MediaSense API と参照しますある特定のプライマリ コールと関連付けられるコールが確認できます。クライアントが CCID <C1> のそのエージェント A によって使用される extn 1000 を、知っていることを仮定して下さい。この準がコールを参照すると見つける手順:

ステップ 1 getSessionByCCID(<C1>) の発行によってプライマリ コールのためのセッション メタデータを取得して下さい。

ステップ 2 sessionStartDate ( コールそれ <Ta> )、および sessionDuration を得て下さい。

ステップ 3 <Ta> へ sessionDuration を追加することによって sessionEndDate ( コールそれ <Tb> ) を計算して下さい。

ステップ 4 この API 要求を実行して下さい:

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

このクエリは複数のセッションを戻すことができます。それが場合、すべては同一コールと関連付けられると仮定することができます。

## 2.4. 複数の参加要素からのコール 検出を参照して下さい

コンサルト コール 検出 セクションで述べられるプロシージャは最初の電話を受信したデバイスからなされるコールを参照するとすべてが見つけます。ただしそこに、何コールを作られますからコールが続いて転送されたデバイス参照しますか。

このプロシージャを考慮して下さい:

1. 発信者はエージェント 1 を呼出します
2. エージェント 1 はエージェント 2 によって、そしてドロップ相談します
3. 発信者はエージェント 2 によって話します
4. エージェント 2 はエージェント 3 によって、そしてドロップ相談します
5. 発信者はエージェント 3 によって話します

このプロセスはエージェント 2 とエージェント 3.間のコンサルト コールをつかまえません。

これが Unified Border Element コールであるので、発信者およびエージェントのそれぞれ間の接続すべてが同じ記録セッションに含まれている、と含まれる同じセッションの参加要素としてエージェントすべてがいつかリストされているというファクトをできますというファクト利用。従って、プライマリ セッション メタデータから、複雑だったすべての deviceRefs のリストを集めることができます。それらのセッションを見つけるために、要求毎に 1 つの deviceRef とともに getSessionByDeviceRef に規定しますプライマリ セッションの時間 範囲を、一連のコールをすることができます。

また、プロセスはこのような単一 getSession 要求と簡素化することができます:

```
{
  "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
    ]
}
]
}
]
```

このクエリはオリジナル プライマリ コールおよび転送すべてと関連付けられるすべての相談コールを戻します。

このプロシージャは実際にネットを余りに広く投げます。たとえば、deviceRef 4000 のエージェントが後 <Ta> 開始することを起こった、そして前に彼が疑わしいコールに追加された完全に独立したコールを行ない、終了したらこのプロシージャは独立したセットで呼出しなさいこと含んでいます。この問題はプライマリ セッションのメタデータで入手可能な情報と、しかし解決することができます。各参加要素の情報は彼が彼の保有のセッションおよび期間に加入したタイムオフセットが含まれています。クライアントコードは上で受け取ったリストから無関係なセッションを単に削除する情報として使用する可能性があります。または、それは正しく各エージェントがプライマリ コールにあった時間をフレーム化する一連の直接 getSession をか getSessionByDeviceRef クエリを作り出す可能性があります。リーダーにこと演習として去ります。

## 2.5. 要約

前のセクションでは、ある特定の Cisco MediaSense 記録セッションと関連付けられたすべてのセッションの検索のための procedures を示しました。ただしある特定のコールが複数のセッションに分けられるかもしれないことが、また中間コール コーデックの場合にはように変更しますわかりました。

どのように発信者の相互対話に接続されるすべてのセッションと関連付けられるすべての記録物を (、また相談します) 取得しますか。

返事によっては倍数の detection のための手順参照しますコールがこの伸ばすことです。最初に、疑わしいプライマリ セッションの CCID を共有するすべてのセッションを集めます。それから、それらのセッションレコードすべてからの参加者一覧表を作成します。次に、最新のセッションの終わりを通して最も早いセッションの sessionStartDate として時間範囲を計算します。最後に、示されている getSession クエリを実行できます。

の前のように、余りにも多くの記録物のキャプチャで終わることができます従ってリストからそれらの無関係なセッションを削除するために後処理ステップを実行する可能性があります。

## 3. どのように他のソリューション コンポーネントの外観と Cisco MediaSense のコールを関連付けますか。

### 3.1. 識別相関表

これら二つの表— Unified Border Element コールの Unified Communications Manager コールのための 1 つおよび 1。各カラムはソリューション コンポーネントを表しますまたは最初のカラムとのプロトコルは、Cisco MediaSense を表します。各行は識別の特定の型を表します。

表を読むために、知っている表し、カラムに水平に表すコールを見つけたいと思うソリューション コンポーネントを検知する データ項目をセルから始めて下さい。そのセルのエントリはどん

な名前を正確のターゲット コンポーネントによって同じデータ項目識別されるか示します。ターゲット コンポーネントにその行で空白セルがある場合、そのデータ項目はそのコンポーネントに知られていません。代わりにそのセルがターゲット コンポーネントのカラムで空白ではない別の行に縦に交差できる介入カラムを探することができます。

たとえば、Unified Communications Manager コールと、GED-188 CallReferenceID を知っている、そして Cisco MediaSense のコールを見つけたいと思うと仮定して下さい。MediaSense カラムで No 値がある、従ってそれに直接マップできませんチェックして下さいことが GED-188 カラムから残、わかります。

ただし、行を渡ってジグザグに進むことができるカラムがあります: Unified Communications Manager CDR。クライアントは IncomingProtocolCallRef が GED-188 CallReferenceID と一致する 1 つを捜すことによって適切な Unified Communications Manager CDR レコードを選択できます。従ってレコードは MediaSense NearEnd xRefCi と同じである、呼出し Cisco MediaSense の対応したレコードを見つけるのに使用することができます値を示すこと destLegCallIdentifier。

Unified Communications Manager CDR レコードは時間までしかし、従ってこの方式がしか歴史的に使用しないことができるコールの終わりが終わった後書き込まれません。

同様にもう一つのパスがあります。GED-188 CallReferenceID から下方に検知して下さい。それは MediaSense の deviceRef フィールドを一致するのにまた AlertingDevice および AnsweringDevice を使用できることとなります。この方式はまたリアルタイムではたります。

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

### 考慮事項:

- Unified Communications Manager コールの記録物では、Cisco MediaSense は UCM から実際 CiscoGUID を受け取りますが、他のソリューション デバイスによってキャプチャされる同じ 1 時ではないです。従って MediaSense はこの値を保存しません。
- エージェントからエージェントへのコールに関しては、TCD.InstrumentPortNumber は宛先エージェントの拡張です。呼出すエージェントの拡張は TCD.ANI で見つけることができます。
- CCID は 10 デイジット 10 進数の 4 組のハイフン分けられたセットの Decimal 形式の

CiscoGUID です。これらはが付いている Hex 形式に 8 桁十六進番号への各 10 デジット 10 進数の改宗者単に変換することができ、ハイフンを取除きます。CiscoGUID は UCCE で使用されるところに、Hex 形式にあります。

## 4. どのトラックがコーリングパーティがあり、どのトラックが被呼加入者があるかどのように判別しますか。

### 4.1. コールに関しては CUBE によって分岐しました

CUBE に関しては 0 を常にマップします固定レグ メディア ストリームに呼出しましたり、トラックします。固定レグはプロファイルを記録するメディアが設定されるダイヤルピアです。第 2 トラック マップ非固定レグ。

受信 dialpeer で有効になる プロファイルを記録するメディアがある場合固定レグは内部レグになります。すなわち、コーリングパーティはトラック 0 に現われ、被呼加入者はトラック 1.に現われます。

送信 dialpeer で有効になる プロファイルを記録するメディアがある場合固定レグはレグになります。それでケースはトラック 1 にコーリングパーティ現われ、被呼加入者はトラック 0 に現われます。

### 4.2. コールに関しては統一された CM 電話によって分岐しました

簡単なコール シナリオで、分岐する統一された CM の場合どのメディアにトラックしなさいかメタデータでどのパーティがあるか判定するのに xRefCi フィールドを使用できます。数値的により小さい xRefCi は通常コーリングパーティのトラックを示します。被呼加入者のトラックは数値的により大きいです (通常 1 つ、しかしそれ多くは適度に付システムの下である可能性があります)。ただし、これらの xRefCi 値はゼロに結局ラップします。このように位置が反転することを 1 つの値は大きい数値であり、他は小さい数であることが分れば、仮定します。

より複雑なシナリオでは、このアルゴリズムは常にはたきません。補助サービスが転送および会議のような、呼び出されれば、および UC マネージャ クラスタが複数のノードで構成されていれば、xRefCi 値は必ずしも次々に生成されないし、順序にまったく意味があると仮定できません。xRefCi 値の特定のペアの発注 シーケンスが信頼される場合があるかどうか判別する直接の方法は xRefCi 値の最初のバイトを検知することです。このバイトはその特定の識別が作成された UC マネージャ ノード ID を表します。2 つの xRefCi 値の最初のバイトが同じである場合、順序は正しいです。それらが異なっている場合、発注は正しくないかもしれません。

これらのケースに関しては、リアルタイムのコールの方向を判別する唯一の方法は JTAPI イベント供給のような他のどの出典からの情報も、得ることです。コールが終了し、数分が経過したら、コールの方向を常に判別し、コールがあるように UC マネージャの CDR データを確認できます。具体的には、CDRレコードの origLegCallIdentifier フィールドは発信者を常に表します。

## 5. CLOSED\_ERROR のセッション状態のための原因とは何か。

CLOSED\_ERROR のセッション状態のための原因は下記のものを含んでいます:

1. 呼出し制御サーバは開いたのか Close 要求のためのメディア (記録) サーバからエラー応答を受け取りました。

2. 呼出し制御サーバは SIP シグナリング エラーを、たとえば抜けた ACK 検出しました。
3. セッションは正常に閉じられましたが、すべてのトラックにゼロ サイズがあります。

セッションが ACTIVE 状態にあるとき、セッションが閉じるまで期間が知られないのでメタデータに期間がないことは正常です。

セッションに関してはセッションかトラック期間フィールドが CLOSED\_ERROR 状態にあるイベントが getSessionData データになれば、そしてこのトラックのためのメディアは利用できません。

## 6. プルーニングされ、削除されたセッション間の違いとは何か

これら二つのクエリを考慮して下さい:

### 6.1. getAllPrunedSessions クエリを使って

このクエリは一組のセッションを、セッション状態の削除されるすべて戻します:

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

### 6.2. getSessionData クエリを使って

このクエリはセッションを戻しません:

```
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. なぜプルーニングされ、削除されたセッションの動作違いか。

動作違いは意図的です。これを MediaSense ドキュメントのセクション参照して下さい：

- API パラメータ説明: `getAllPrunedSessions` API 説明:

すべてのプルーニングされた記録物を検索するのにこの API を使用して下さい...プルーニングされる条件が記録物を示す Cisco MediaSense システムによって削除される。 `deleteSessions` API を使用して明示的に記録を削除する場合、これらはプルーニングされた記録物として削除された記録物考慮されません。

- セクション 予防的な 記憶域管理の下の MediaSense SRND:

セッションがプルーニングされる時、メタデータはデータベースにこれらのセッションと関連付けられるこれらのセッションが「プルーニングされる」としてマークされた後でさえも、残ります。このメタデータ記録物と比較される多量の記憶域を自身は奪取しませんが、領域を奪取し、定期的に取り除く必要があります。このアクティビティを援助するために、クライアントは定期的にプルーニングされたセッションのための API 要求を発行するかもしれませんがセッションによってプルーニングされるイベントを受け取り、明示的にクライアントがもはや必要としないこれらのイベントを削除するためにまたはクライアントは選ぶかもしれません。

明白になるために、2つのクエリは全く異なります。実際、第2クエリは ( `lwhich` は状態が削除されるすべてのセッションを検索します ) 空のセットを常に戻します。正常な日常のクエリはそれが要求されることがでも、削除された状態とのセッションをフィルタ・アウトします。唯一の例外は `getAllPrunedSessions` です。この例外はアプリケーションがこれらのセッションは削除されるように要求できるようにプルーニングされたセッションを見つけるためにアプリケーションを助けるように意図されています。

`getAllPrunedSessions` から得るプルーニングされたセッションのリストの `deleteSessions` API を使用すれば、これらのセッションは `getAllPrunedSessions` の結果にもはや現われません。そのようなセッションはメタデータから完全にすぐに取除かれます。

これを検知するもう一つの方法はプルーニングされたセッションが削除されたセッションと同じ事柄ではないことです：

1. プルーニングされたセッションは MediaSense システムのアルゴリズムによって削除のために示されました。人はこれらのセッションをプルーニングするデジジョンに関連しませんでした。従ってこれらのセッションが削除された状態に変えられるのに、これらのセッションはメタデータから実際に取除かれません。人間の ( またはアプリケーション ) 介入が必要となります。これらのセッションが削除された状態にあるので、これらのセッションはほとんどのクエリに目に見えませんが、これらのセッションは `getAllPrunedSessions` クエリ API に目に見えます。またどの mp4 ファイルでもこれらのセッションのために生成されたら、これらの mp4 ファイルはプルーニングされたセッションが実際に削除されるまでディスクにありおよびダウンロードのために利用可能であり続け続けます。
2. 削除されたセッションは明示的に `deleteSessions` API を呼出すことによって示されます。このマーキングはセッションに既にプルーニングされているまたはまだ削除されていないセッションにすることができます。セッションが `deleteSessions` API によって削除されたら、このセッションはあらゆるクエリにもはや目に見えませんが、これには



getAllPrunedSessions API が含まれています。これらはメタデータから削除されたセッション ディスク領域が再利用することができるようにすぐに取除かれます。

## 7. メディア分岐のための TDM ゲートウェイの設定方法か。

フローおよびあなたを呼出す PSTN ゲートウェイがあるときそれらのコールを記録したいと思って下さい。これらのコールは TDM に SIP コールです。ただし、メディア分岐は一口に SIP コールだけで利用可能です。

これらのコールは記録することができます。これらのコールはルータを通して指示されてできます二回目。設定指導および他の詳細は [この白書](#)で見つけることができます。

## 8. ハントする グループを使用する場合の実際の宛先電話をキャプチャする方法か。

CUBE から分岐するメディアを使用するとき MediaSense メタデータは普通被呼加入者の拡張が含まれています。ただし呼出される数が通信マネージャ ハントする グループ パイロット数なら、そしてデフォルトでメタデータはことだけパイロット数含んでいます。それは実際にコールに回答した電話の拡張が含まれていません。

これを変更できる通信マネージャ設定があります。ハントする/パイロット設定 ページで、セクションを接続された当事者変形と資格を与えられる探して下さい。接続されたパーティとして設定表示行グループ メンバー DN は始動させる必要があります。

この機能は通信マネージャ 9.0(1) および以降で利用できます。

## 9. メカニズムが Unified Communications Manager ネットワークベース記録好まれた分岐として推奨されてある理由

Unified Communications Manager ネットワークベース記録 ( NBR ) を使うと、コールを記録するのにゲートウェイを使用できます。NBR は Unified Communications Manager がデバイス、場所、または地理学に関係なく記録コールを、ルーティングすることを可能にします。NBR を使うと、コール記録メディアは IP 電話または SIP トランク上の Unified Communications Manager に接続されるゲートウェイからソースをたどることができます。Unified Communications Manager は動的にコールフローおよびコール参加要素に基づいて右のメディア ソースを選択します。

NBR は別途の記録設定が必要とならないので統合サービス ルータ ( ISR ) が利用できないとき自動フォールバック構築ブリッジ ( 胸当て ) を提供します。これは Unified Border Element によって参照するコールが記録できない、従って胸当てが別々に有効になる必要があるので顧客によって参照すれば記録ポリシーのコールがエージェント エージェントを含みたいと思えば役立ちます。

NBR および胸当てコールは両方 Unified Communications Manager JTAPI から利用可能である xRefci を使用して関連させることができます。意味する CTI サーバも CTIOS 接続も必要とならない CISCO-GUID は必要ではありません。単一相関識別があるので、コンポーネントを渡す相関はより強く、コールフローの均一方法依存しないですることができます。

直接ダイヤルされて NBR が、またダイヤラ始められた送信呼び出しは他のソリューション コン

ポーントの外観に関連させることができます。

NBR を使うと、TDM ゲートウェイ記録はルータのキャパシティの分割無しで自動的に使用されます。現在、TDM ゲートウェイ記録は MediaSense 10.5 でサポートされません。

## 10. ノードはなぜ MediaSense 10.5 にアップグレードするために時間がかかりますか。

ノードは保持する記録物の数およびサイズによってアップグレードするために決まります数時間がかかる場合があります。MediaSense 10.5 に関しては、非常に大きいデータセットのノードをアップグレードするとき、1,000,000 記録物ごとのおよそ 90 の追加分がかかります。

## 11. MediaSense 検索および演劇アプリケーションのロシア時間帯変更の影響とは何か。

MediaSense のユーザは検索し、影響を与えられた時間帯の何れかにいるか、または検索基準の影響を与えられた時間帯を選択すれば演劇アプリケーションは影響を受けています。MediaSense とインターフェイス接続するサードパーティパートナー製品はそれぞれ時間帯表をアップデートするまで同様に影響を受けます。

回避策は都市がもはや正しくなくて GMT からの正しいオフセットと一致する時間帯を選択することです。

## 12. MediaSense によってサポートされる言語とは何か。

MediaSense によってサポートされる言語はここにあります:

- アラビア
- デンマーク
- オランダ語
- 英語 ( 米国 )
- フィンランド
- フランス語
- ドイツ語
- イタリア語
- 日本語
- 韓国語
- ノルウェー語
- ポーランド語
- ポルトガル語 ( ブラジル )
- ロシア語
- 簡体字中国語
- スペイン語
- スウェーデン語
- 繁体字中国語
- トルコ語

## 13. MediaSense システムパフォーマンスを監視する方法か。

MediaSense システムパフォーマンスを監視するために、RTMT ツールまたは Cisco Prime Collaboration 保証ツールが付いているこれらの主要業績評価指標 (KPIs) の値を分析して下さい。

RTMT ツールまたは Cisco Prime Collaboration 保証ツールに関する詳細については、[Cisco MediaSense ユーザガイド](#)の統一された RTMT 管理および Cisco Prime Collaboration 保証管理セクションを参照して下さい。

主要業績評価指標	KPIs および閾値	
	RTMT カウンター	推奨される 閾値
コール成功率	MediaSense 呼制御サービス > セッション レコード番号エラーなしで	99.99 %
API のための API 応答平均時間	Cisco MediaSense API サービス > 平均クエリの応答時間	60 秒
記録開始する平均設定遅延	Cisco MediaSense 呼制御サービス > 平均設定遅延	3 秒
CPU 平均利用	プロセッサ > CPU タイム % の	90 %
メモリは利用を意味します	使用されるメモリ > %Mem 廃棄されるネットワーク インターフェース > Rx > eth0	70%
RTP/UDP パケット ドロップする	ネットワーク インターフェース > Rx エラー > eth0	0

## 14. ブラウザを MediaSense の内部ブラウザ プレイヤーを実行するために設定する方法か。

ブラウザに基づいて、内部ブラウザ プレイヤーを実行するためにこれらのステップを実行して下さい:

### Internet Explorer 9

1. MediaSense 検索および演劇で、記録セッションの Play アイコンをクリックして下さい。

### Internet Explorer 11

前提条件: MediaSense 11.0 の新しいインストールの場合には、MediaSense ノードがそれぞれ完全修飾ドメイン名 ( FQDN ) を使用してクラスタに追加されるようにして下さい。  
MediaSense 11.0 へのアップグレードの場合には、ホスト名を使用して以前に追加された MediaSense ノードがそれぞれ FQDN によって、今表示する必要があるようにして下さい。

### Mozilla Firefox

1. Mozilla Firefox 信頼されたサイトの mp4url のポート 8446 のための MediaSense ノードの自己署名証明書を追加して下さい。

## MediaSense サーバコンフィギュレーション ウィンドウ ( Cisco

MediaSense Administration > システム > MediaSense サーバコンフィギュレーション ) の MediaSense Server リストをチェックして下さい。

次の操作を行ってください。

1. MediaSense に FQDN だけを使用して `mediaurls` の `mp4url` および他を準備させます「本当」として一定 `hostnameformediaurl` CLI を設定して下さい。

```
admin:set useHostNameForMediaURL
admin:set useHostNameForMediaURL true
```

2. プロパティをアクティブにするために構成 サービスを再開して下さい。

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

注: サービスがきちんと再起動しない場合、同じコマンドを再度実行して下さい。

3. 構成 サービスが再起動した後、署名し、MediaSense 検索および演劇に署名して下さい。

制限: MediaSense ノードが IP を使用して以前に追加されたら、そしてノードは MediaSense 11.0 にアップグレードの後でさえもだけ IP によって表示され続けます。内部ブラウザプレーヤーは `hostnameformediaurl` CLI コマンドの値に関係なく Internet Explorer 11 に、取り組みません。この場合 `hostnameformediaurl` CLI コマンドが「本当として」。設定すべきではないことが、推奨されます Windows によって信頼される権限に MediaSense 自己署名証明書を追加するために次のステップを実行して下さい。

1. MediaSense 検索および演劇を開いて下さい。

機密保護証明書ポップアップウィンドウは現われます。

2. 『Continue』をクリックして下さい。

MediaSense 検索および Play ウィンドウは現われます。

3. アドレスバーで、Certificate エラーアイコンをクリックして下さい。

2. 証明書を信頼するために『Yes』をクリックして下さい。

注: 提供される自己署名証明書が目標とされた MediaSense ノード 証明書の技術的詳細ことをの FQDN の検証によってであることを確認して下さい。

3. 指定記録に相当して Play アイコンをクリックして下さい。

内部ブラウザプレーヤーは指定記録セッションをします。

2. 自己署名証明書を追加するに、記録セッションの Downl アイコンをクリックし、mp4 択して下さい。

この接続は信頼できないポップアップウィンドウ現われますで

2. 自己署名証明書を追加するに、記録セッションの Downl アイコンをクリックし、mp4 択して下さい。

この接続は信頼できないポップアップウィンドウ現われますで

注: 提供される自己署名証明書 標とされた MediaSense ノー 明書の技術的詳細ことをの F の検証によってであることを して下さい。

4. 証明書を『View』をクリックして下さい。  
証明書ポップアップ ウィンドウは現われます。
5. 『install certificate』をクリックして下さい。  
Certificate import ウィザードは現われます。
6. 『Next』をクリックして下さい。
7. 証明書ストア ウィンドウで、**Place all certificates in the following store** オプション・ボタンを選択し、『Browse』をクリックして下さい。  
『Certificate』を選択 ストア ダイアログ ボックスは現われます。
8. **Show physical** ストア チェックボックスをチェックし、**信頼されたルート Certificationhorities** フォルダを選択して下さい。
9. 次に『OK』をクリックすれば。
10. certificate import を完了するために『Finish』をクリックして下さい。  
。ポップアップ ウィンドウに警告するセキュリティは証明書のインストールを確認するようです。
11. 『Yes』をクリックして下さい。  
次のメッセージが現れます。

The import was successful.

12. [OK] をクリックします。
13. 証明書ポップアップ ウィンドウで『OK』をクリックして下さい。
14. 終わりはブラウザを開き。
15. 開いた **MediaSense 検索および演劇**。  
機密保護証明書 issxvi はまだ持続します。
16. 証明書 アドレス ミスマッチ チェック boxviiunder セキュリティについての警告を Tools > Internet Options > Advanced の順に進んで下さい、チェックを外して下さい。
17. [Apply]、次に [OK] をクリックします。
18. ブラウザおよび開いた **MediaSense 検索および演劇再起動**して下さい。

MediaSense サーバがブラウザの FQDN によってアクセス可能であることを確認して下さい。そうでなかったら、

C:\Windows\System32\drivers\etc にナビゲートし、Notepad のホストファイルを開き、ファイルの下部ので MediaSense サーバおよび FQDN の IP アドレスを追加して下さい。内部ブラウザプレイヤーは Internet Explorer 11.に取り組み始めます。

**注:** 記録がクラスタの別の MediaSense ノードにある場合、信頼された権限のその MediaSense ノードの証明書を追加するためにプロンプト表示されます。

3. **理解します危険性リンク** をクリックして下さい。
  4. [Add Exception] をクリックす。
  - 追加セキュリティ例外** ポップアップウィンドウは現われます。
  5. **確認しますセキュリティ例** をクリックして下さい。
- 8446 ポートの特定の MS ノード自己署名証明書はブラウザの信頼された権限に追加されます。