UCCXのJTAPIアーキテクチャとコールフローについて

内容

はじめに

このドキュメントでは、JTAPIの基本機能、そのアーキテクチャ、およびUCCXに関するコールフローについて説明します。

前提条件

要件

次のツールと機能に関する知識があることが推奨されます。

- JTAPI:Java Telephony API（ジャバテレフォニーAPI）
- API：アプリケーションプログラミングインターフェイス

- UCCX:Unified Contact Center Express
- CUCM:Cisco Unified Communications Manager
- CTI:Computer Telephony Integration（コンピュータテレフォニーインテグレーション）

次の項目に関する知識があることが推奨されます。

- Cisco Unified Communications Manager(CUCM)の設定
- Unified Contact Center Express(UCCX)の設定

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアのバージョンに基づいています。

- UCCX バージョン 12.5.1.11002-481
- CUCM バージョン 12.5.1.11900-146

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

このドキュメントでは、JTAPIアーキテクチャ、コールフロー、デバッグの有効化、およびログの収集について説明します。

JTAPIの概要

Cisco Unified JTAPIは、Javaベースのコンピュータテレフォニーアプリケーション用にSun Microsystemsが開発したプログラミングインターフェイス規格として機能します。Cisco JTAPIは、Sun JTAPI 1.2仕様を追加のシスコ拡張機能とともに実装します。Cisco JTAPIを使用して、次のようなアプリケーションを開発する必要があります。

- Cisco Unified Communications Manager電話機を制御および監視します。
- コンピュータテレフォニーインテグレーション(CTI)ポートとルートポイント（仮想デバイス）を使用してコールをルーティングします。

JTAPIおよびUCCX

コンタクトセンターでは、Cisco Unified JTAPIを使用してデバイスのステータスを監視し、適切なタイミングで適切な場所にコールを送信するためのルーティング指示を発行します。さらに、JTAPIを使用して、分析用のコール統計を取得する際の記録指示の開始と停止、CRMアプリケーションへのコールのスクリーンポップ、自動スクリプト、およびリモートコール制御を行うことができます。

JTAPIアーキテクチャ

エンタープライズ環境で使用されるCisco Unified JTAPIは、ユーザのアベイラビリティ、ロケーション、およびプリファレンスを組み合わせて、プレゼンスベースのルーティング用に独自に調整された環境を実現します。

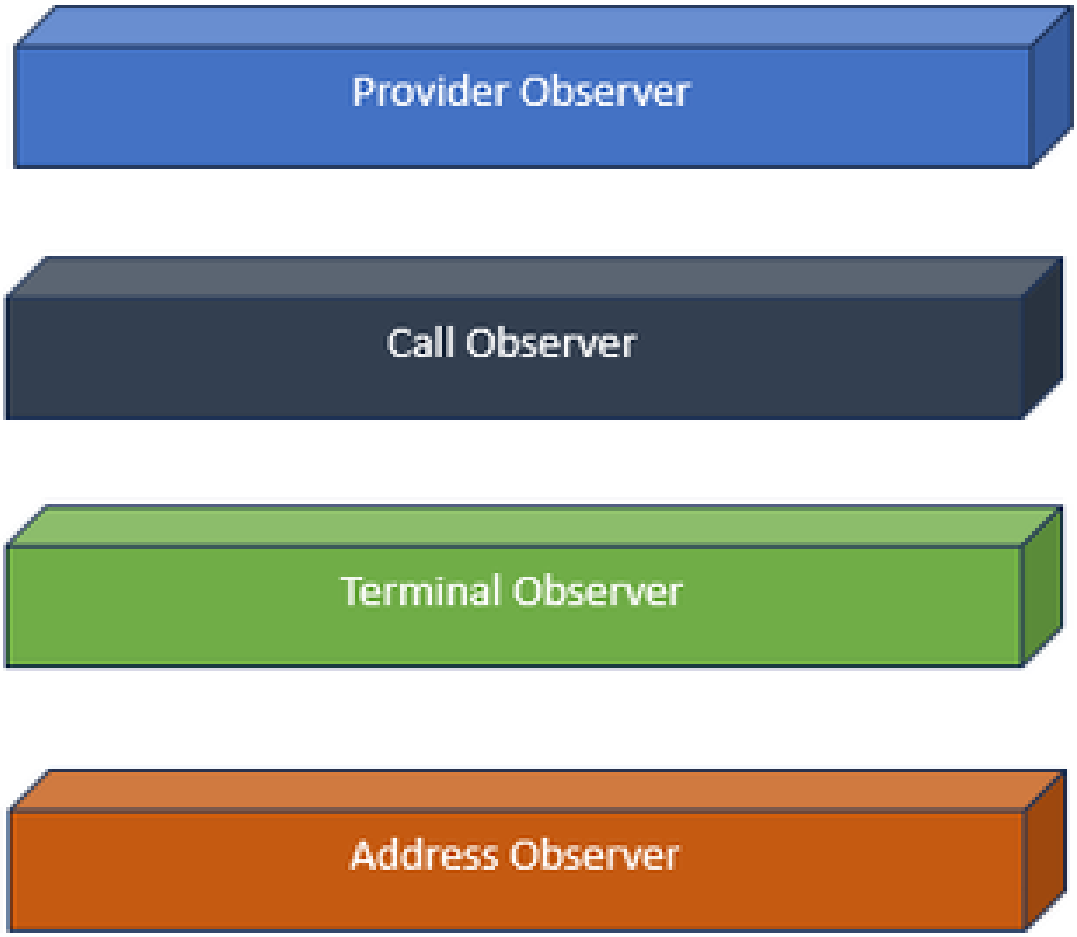
JTAPIのアプリケーションを次に示します。

- JTAPIは、2つ以上の電話と回線の組み合わせをモニタしたり、通知を受けたりすることができます。
- コンタクトセンターアプリケーションは、JTAPIのフルコールモデルを使用します。
- JTAPIは次の機能を提供します。
    - コール制御
    - メディア制御
    - メディア ネゴシエーション

JTAPIオブザーバモデル

次の図は、JTAPIが動作するProvider-Observerモデルを示しています。

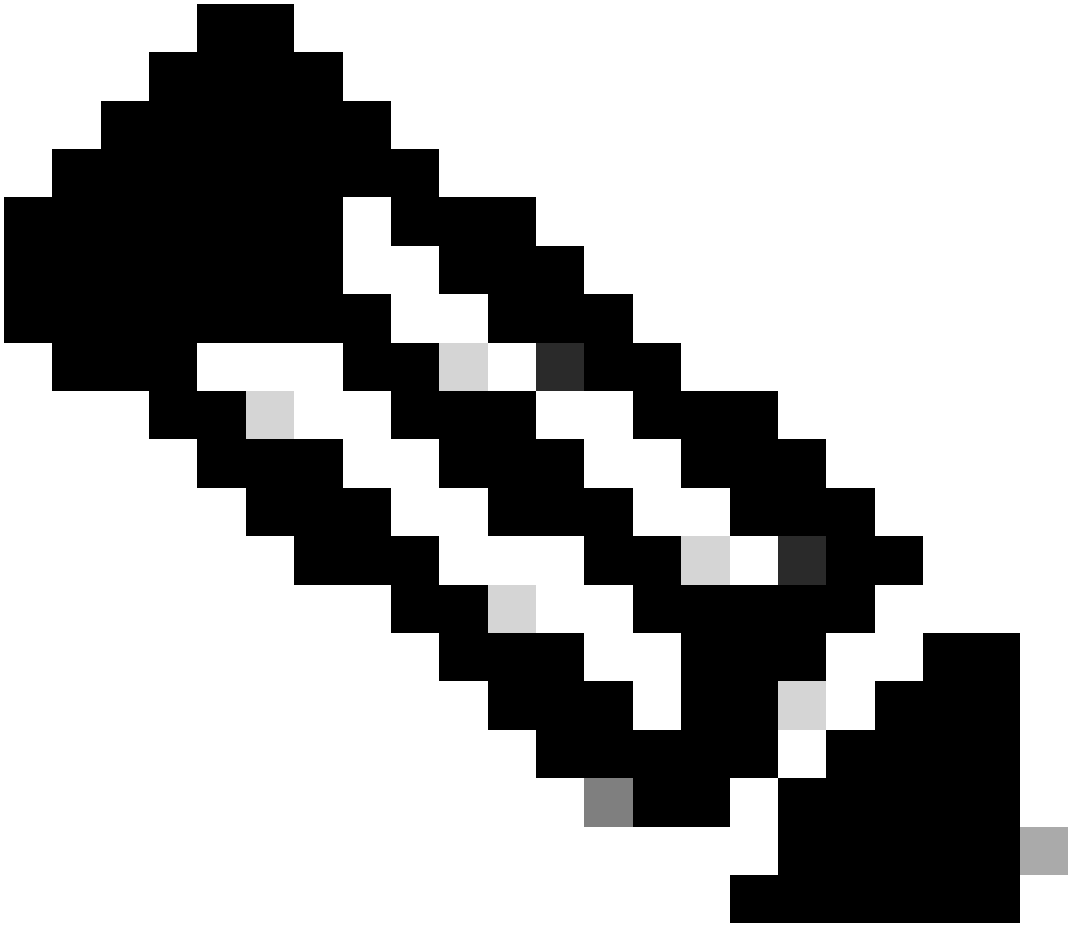
オブザーバ



オブザーバインターフェイス

JTAPIはオブザーバーのアイデアに基づいています。オブザーバーによって、それは出来事を観察する人の考えを指します。複数のオブザーバをシステム内の異なるオブジェクトに配置できます。JTAPIはオブザーバを使用して、オブジェクトの状態変化を認識します。

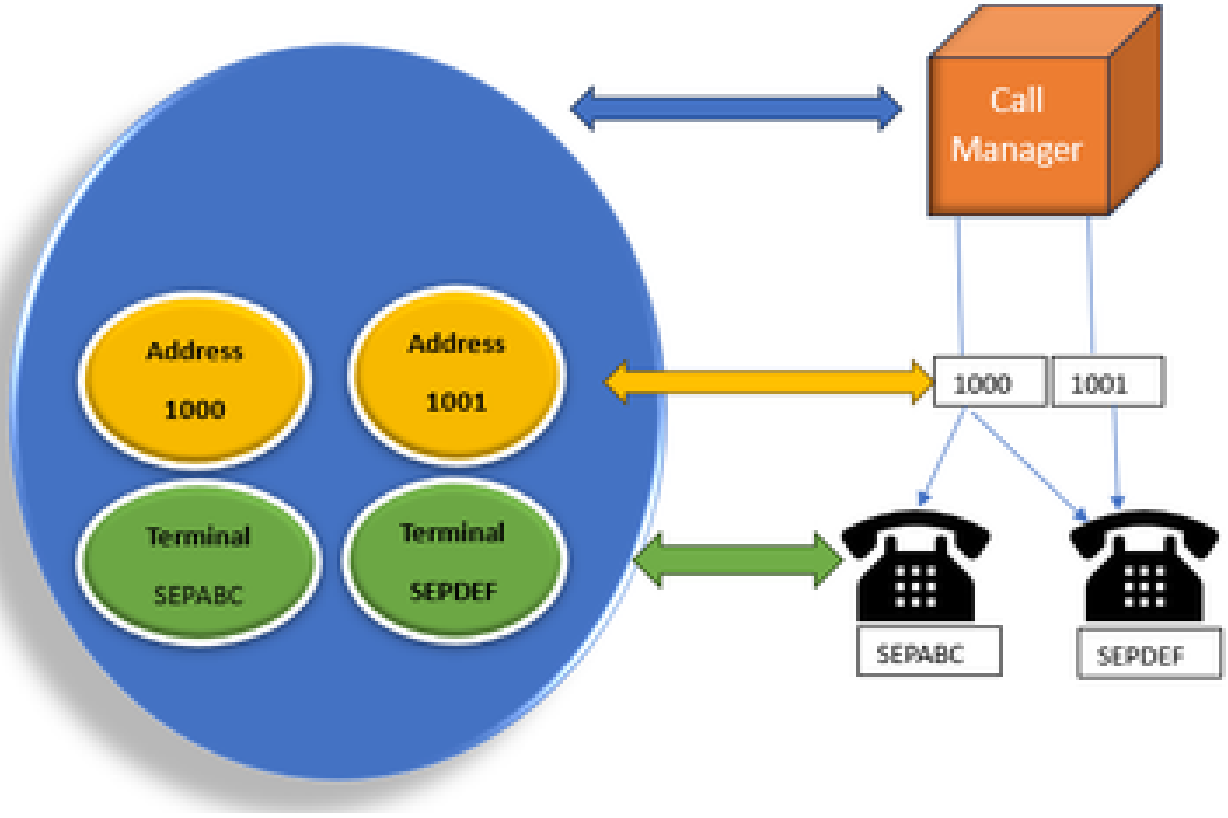
たとえば、オブザーバを回線、電話、端末、アドレスなどに配置し、その状態の変化を学習できます。



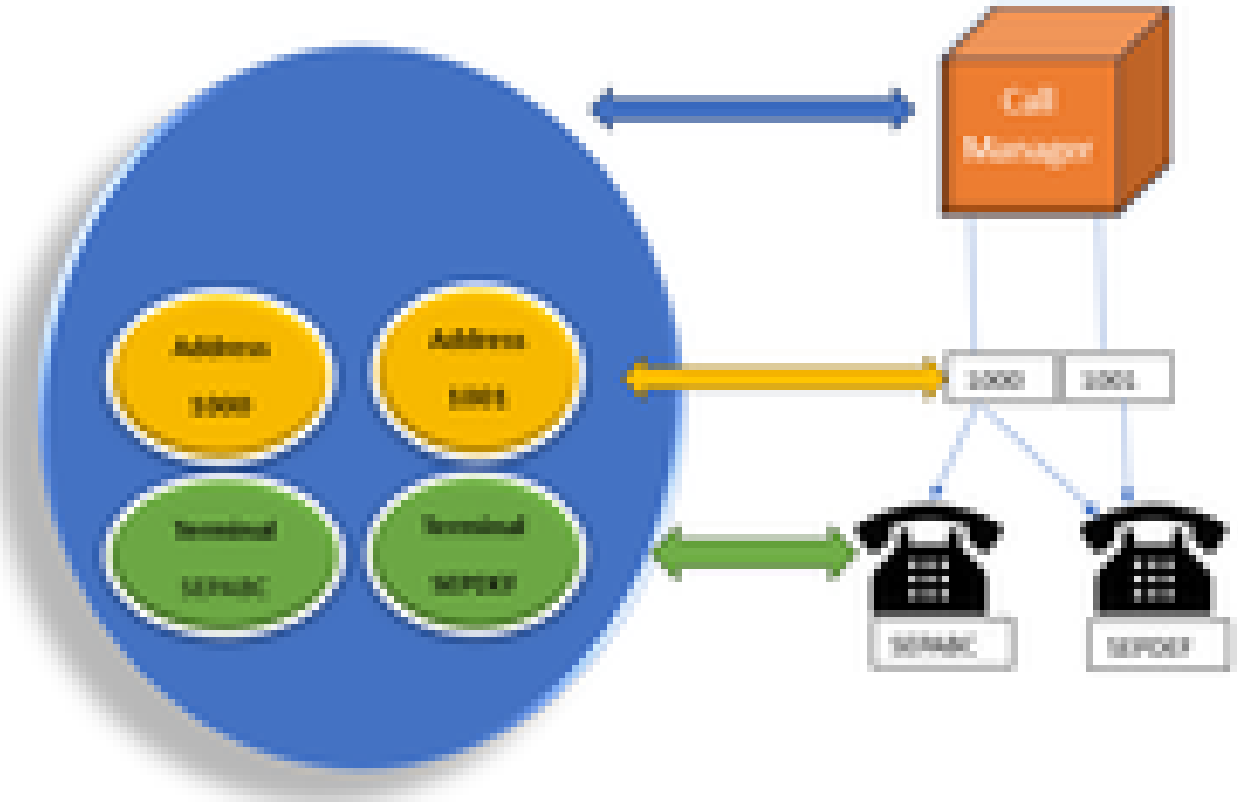
注記：オブジェクトにオブザーバを配置していない場合、オブジェクトに対して実行されたアクションに関する通知を受け取ることはできません。

JTAPIプロバイダーモデル

次の図は、JTAPIが動作するプロバイダーモデルを示しています。



*JTAPI*プロバイダー

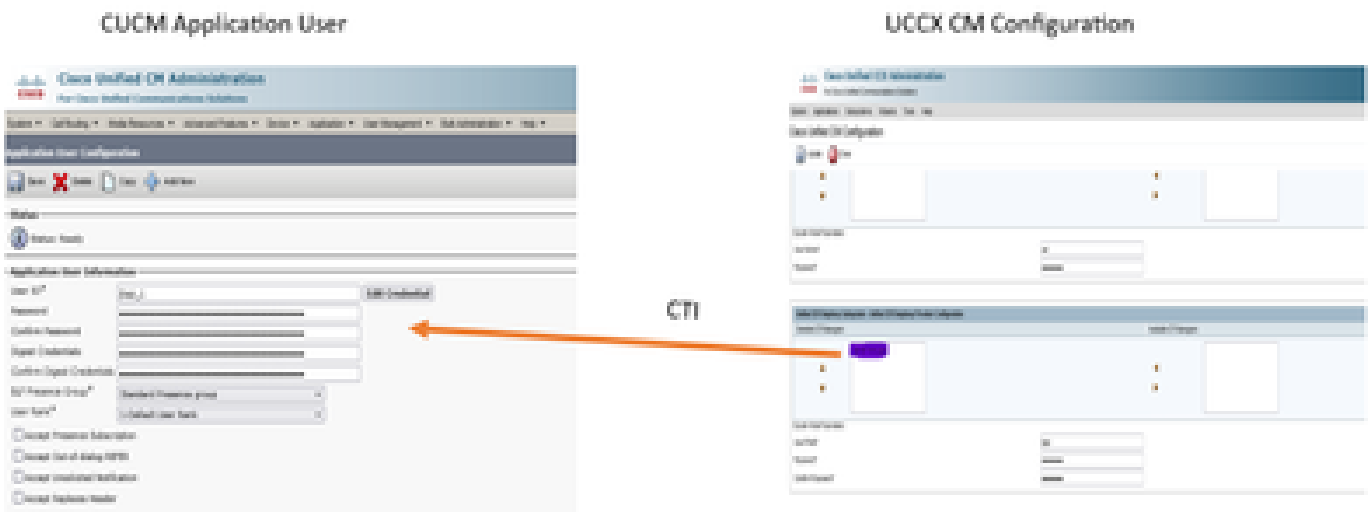


モ
デルプロバイダーモデル

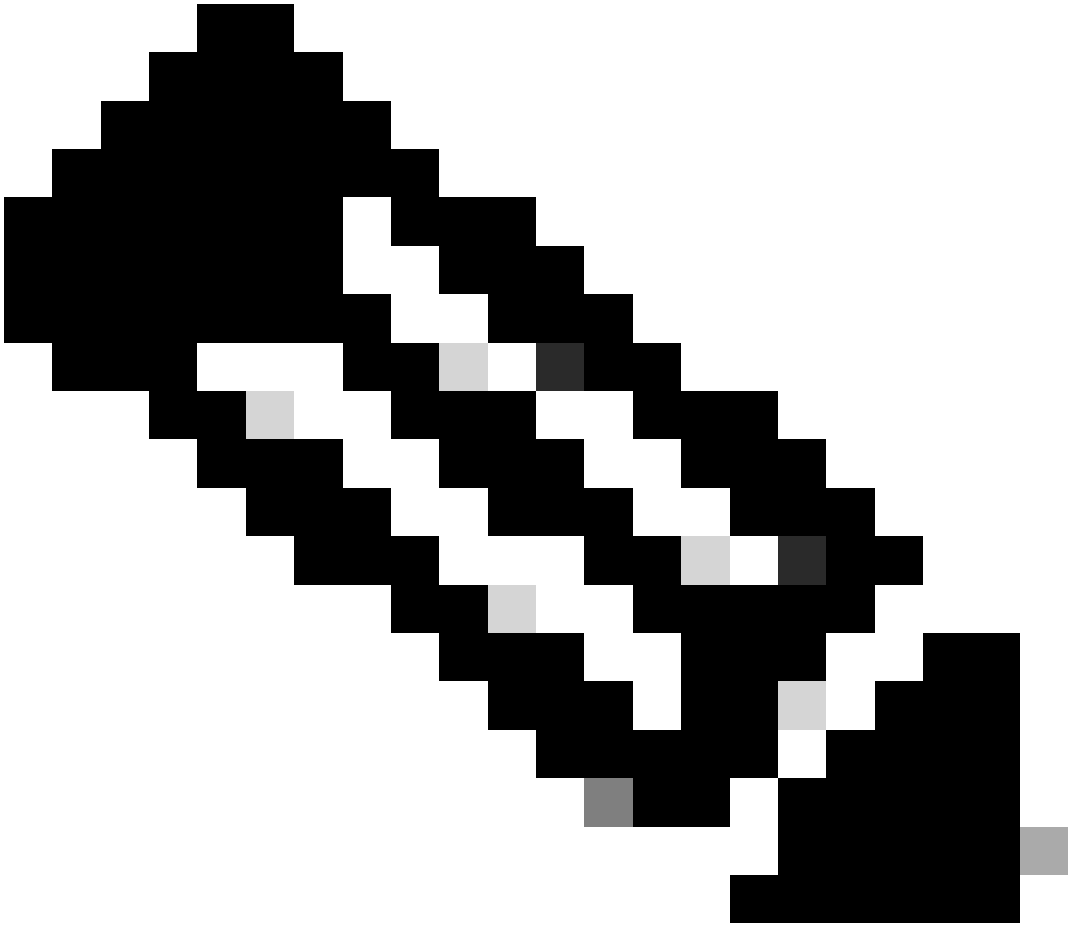
JTAPIプロバイダーは、アプリケーションからCall ManagerまたはCTI Managerへの接続です。プロバイダは、オブジェクトにオブザーバをアタッチするために使用されます。オブザーバをプロバイダに接続することもできます。プロバイダは、オブジェクトに対して実行されたアクションに関する通知を受け取るために使用されます。(オブザーバーが接続されているデバイス（オブザーバーを接続しているプロバイダー）を制御することもできます)。

JTAPI ユーザ

次のスクリーンショットは、JTAPIアプリケーションユーザについて説明するCUCM（左）およびUCCX（右）のものです。



- アプリケーションを起動してCTIマネージャに接続するには、プロバイダーを開きます。
- プロバイダーをオープンする理由は、デバイス、端末などに対するアクションを監視できるようにするためです。
- CUCMでの実装方法は、アプリケーションユーザを通じて行われます。
- このユーザを作成してクレデンシャルを指定すると、最初にCTIを介してCUCMに対して認証が行われます。
- したがって、CUCMで作成されるJTAPIアプリケーションユーザがプロバイダーになります。
- 単なるモニタリングとは別に、デバイスが**Associated Devices**の下にあることを確認して、デバイスを制御できることを確認する必要があります。



注:JTAPIユーザはcucm上に作成しません。これは、CUCM上のAXLを介してアプリケーション(UCCX)によって作成されます。

P1およびP2ハンドル

P1とP2はプロバイダーハンドルです。これらは、同じアプリケーションから複数のプロバイダーを開く場合に重要になります。UCCXから、2つのプロバイダーを作成します。1つはCTIポートとルートポイントを制御し、もう1つはRM-JTAPIプロバイダーと呼ばれるエージェント電話を制御します。UCCXでは、CTIポートとルートポイントを最初に制御するプロバイダー (JTAPI P1プロバイダー) を作成します。P1の後に作成する他のプロバイダは、エージェントデバイスを処理するP2プロバイダまたはRmCmプロバイダです。

P1の新しいコールイベントが表示された場合は、それがCTIポートまたはCTIルートポイントからのコールイベントであることが

わかります。P2の新しいコールイベントが表示されていれば、これは実際の電話機からのコールイベントであることがわかります。CCX側に1つのRM-JTAPIユーザと1つのJTAPIユーザを作成しますが、CUCM側には2つのJTAPIユーザが作成されたことが表示されます。これは、各CCXノードが独自のJTAPIユーザを取得しますが、1つのRM-JTAPIユーザを共有するためです。UCCXでトリガーを作成すると、両方のJTAPIユーザに追加されます。両方のノードがプロバイダを個別に開きます。したがって、ルートポイントで実行されたアクションは、両方のCCXノードで通知されます。

それ以外の場合、セカンダリノードはセカンダリノードであることを通知し続けます。各ノードには、CTIポートの個別のグループがあります。ノード1のCTIポートは、jtapi_1によって制御されます。ノード2のCTIポートは、jtapi_2によって制御されます。

コールがCTIルートポイントに到着すると、両方のCCXノードに新しいコールイベントが通知されますが、アクティブなCCXノードは、自身が制御するCTIポートの1つに対するリダイレクト要求でCall Managerに応答します。

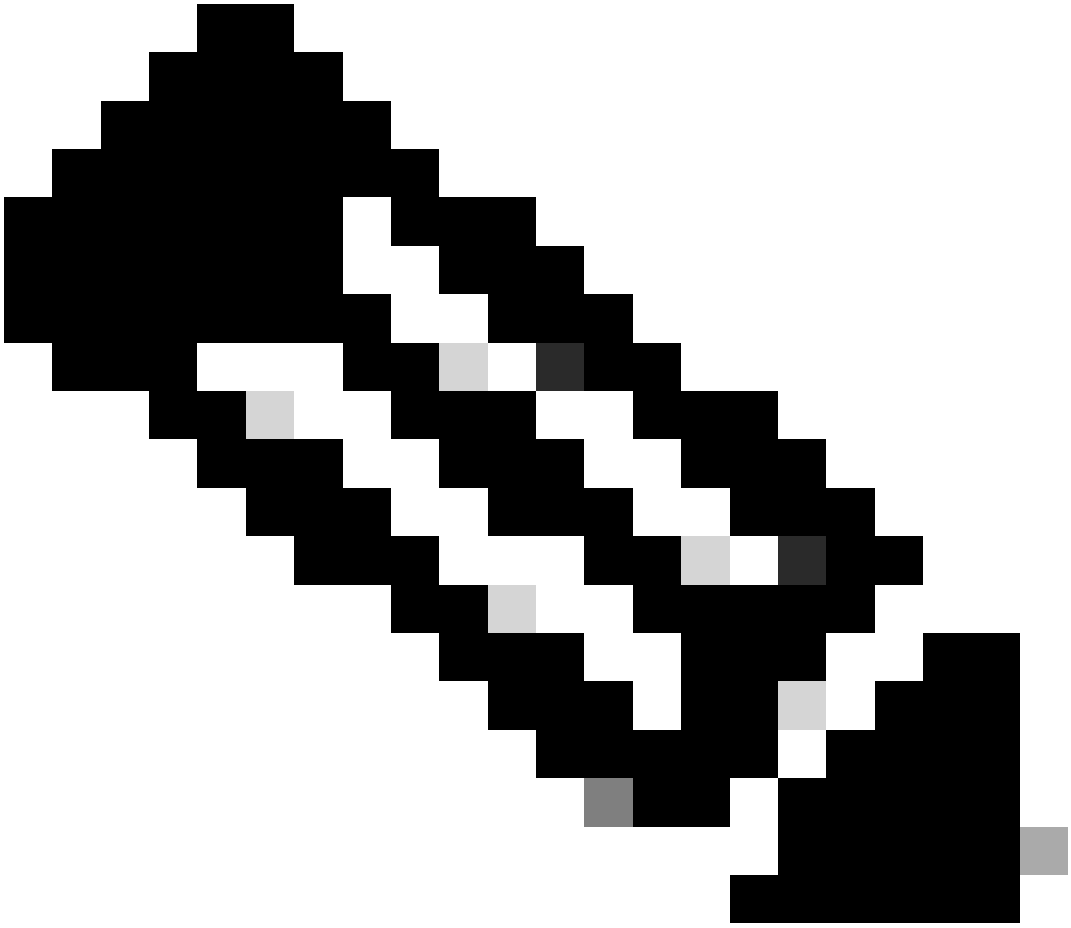
CUCMでCTIルートポイントに対するIPが表示される場合、これはCCXのIPの1つですが、特定のノードがコールをルーティングしているわけではありません。

一般的に、JTAPIユーザからデバイスの関連付けを解除し、再度追加します。この理由は、関連付けを解除すると、プロバイダはその通知を受け取り、それに対するすべての接続を削除します。その後、オブザーバを再度追加すると、オブザーバは再び追加され、プロバイダはopen provider requestを使用してもう一度監視を開始します。制御デバイスリストに追加されるデバイス以外に、個々のデバイス上の**Allow Control of Device from CTI**チェックボックスと呼ばれる別の項目があることがわかります。これは、より詳細なレベルを実現するためです。したがって、ルートポイントが制御リストに追加されていても、CTIチェックボックスがオンになっていない場合は、そのルートポイント上のイベントについて通知を受け取ることはできますが、コール制御でのアクションはできません。

コールフロー

UCCXコールフローに関連するイベントのシーケンスを次に示します。

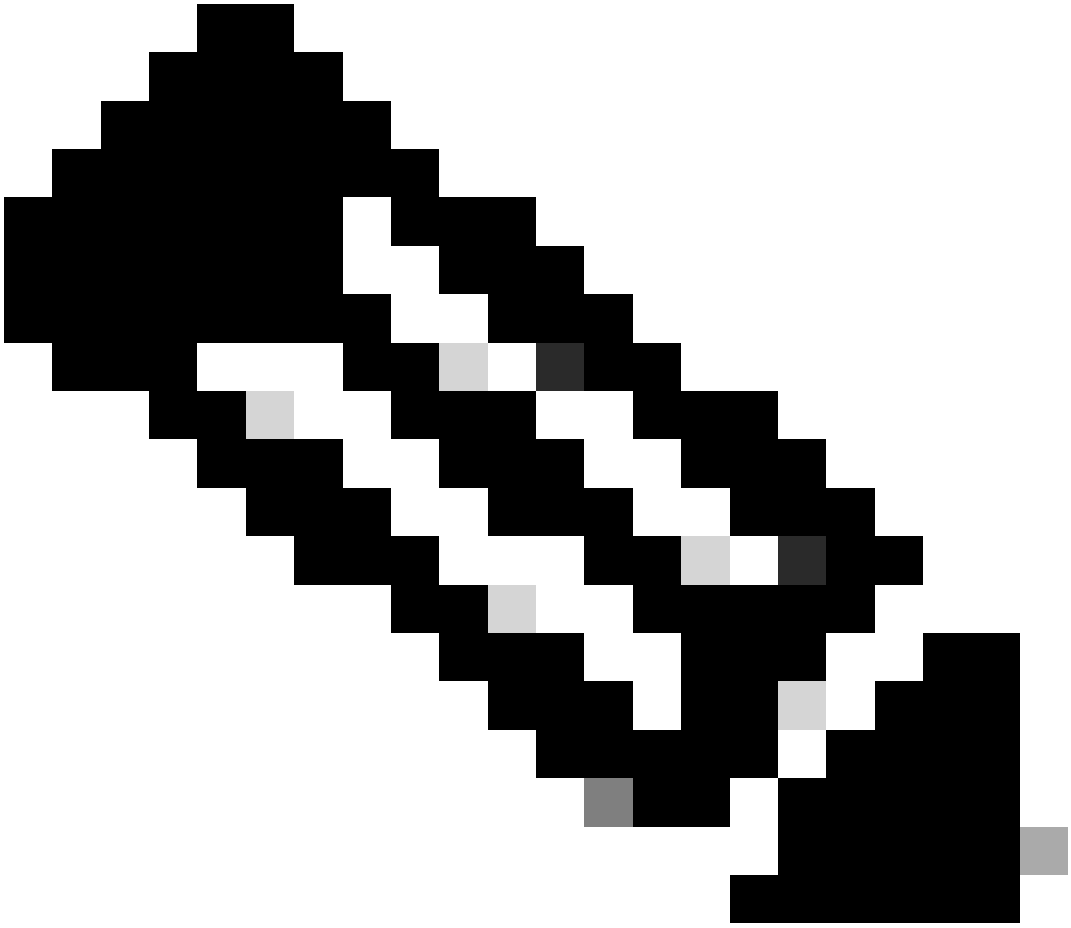
- コールがCTIルートポイントに到着すると、コールはCTIポートにリダイレクトされます。
- CTIポートはuccxボックスのipvmsドライバでメディアチャネルを開きます。
- エージェントがコールを受ける必要があると判断すると、ポートからエージェントへの打診転送が行われます。
- 新しいコールイベントがCTIルートポイントに送信されます。
- リダイレクト要求がCTIポートに送られます。
- コールIDの状態がアイドル状態になります。
- その後、CTIポートへのコールに対して別の新しいコールイベントが発生します。
- リダイレクト応答が0の場合は良好な応答であり、0以外の場合は失敗したことを意味します。
- 次に、call accept要求を送信します（これは応答されず、単にポートで受け入れられます）。
- 次に、スクリプトでステップヒットを受け入れます。これは、コール応答要求がJtapiに着信したときです。



注：これは非常に高速で発生し、Cisco Unity ConnectionからのコールやCUCMからの転送に時間がかかるような複数のイベントが同時に発生する場合もあります。これがacceptステップでRACE状態を引き起こす可能性があるため、この速度を下げます。これを行うには、acceptステップの前にdelayステップを追加します。

11. 次に、ポートから応答が返されます。

12. コール状態が接続済みに変わります。



注：CTIは、新しい要求を送信する前に応答を待つsipまたはskinny電話とは異なり、非同期です。そのため、JTAPI CTIメッセージ内のメッセージの順序が混乱することがあります。

13. ポートから応答応答を受信した後、メディアネゴシエーションが発生します。

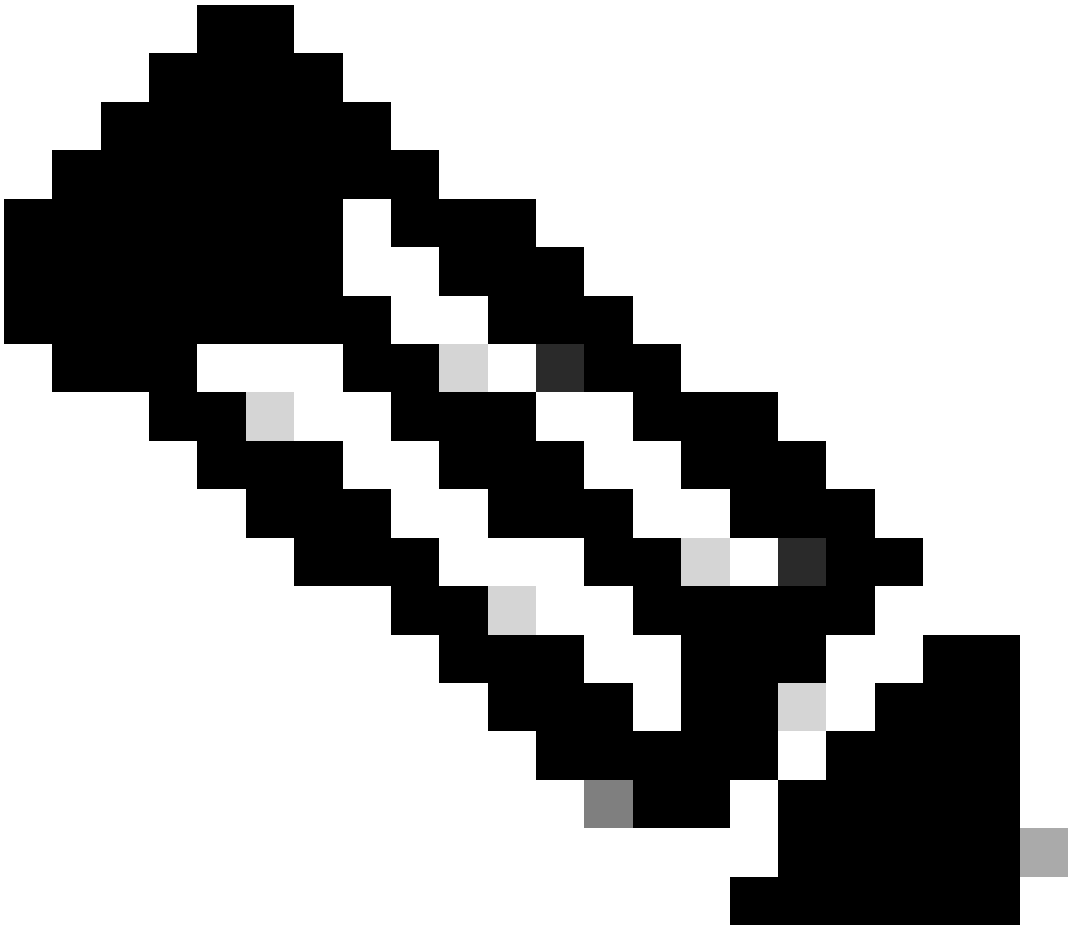
14. ポートはopen_logical_channel要求を送信します。この要求では、ポートと、リモート側にRTPの送信先として指定するIPを共有します。

15. 論理チャネルを開く前に、UCCXボックスのIPVMSドライバとの接続を作成し、RTPストリームを開きます。

16. 次のイベントはstart_receptionイベントで、遠端情報（つまり、発信側デバイスのIPとポート）を通知します。

17. その後、他のメディア信号と同様にstart_transmissionイベントを取得します。

18. これで、IVRとプロンプトが再生されます。



注：ここでコールセットアップが完了します。

---

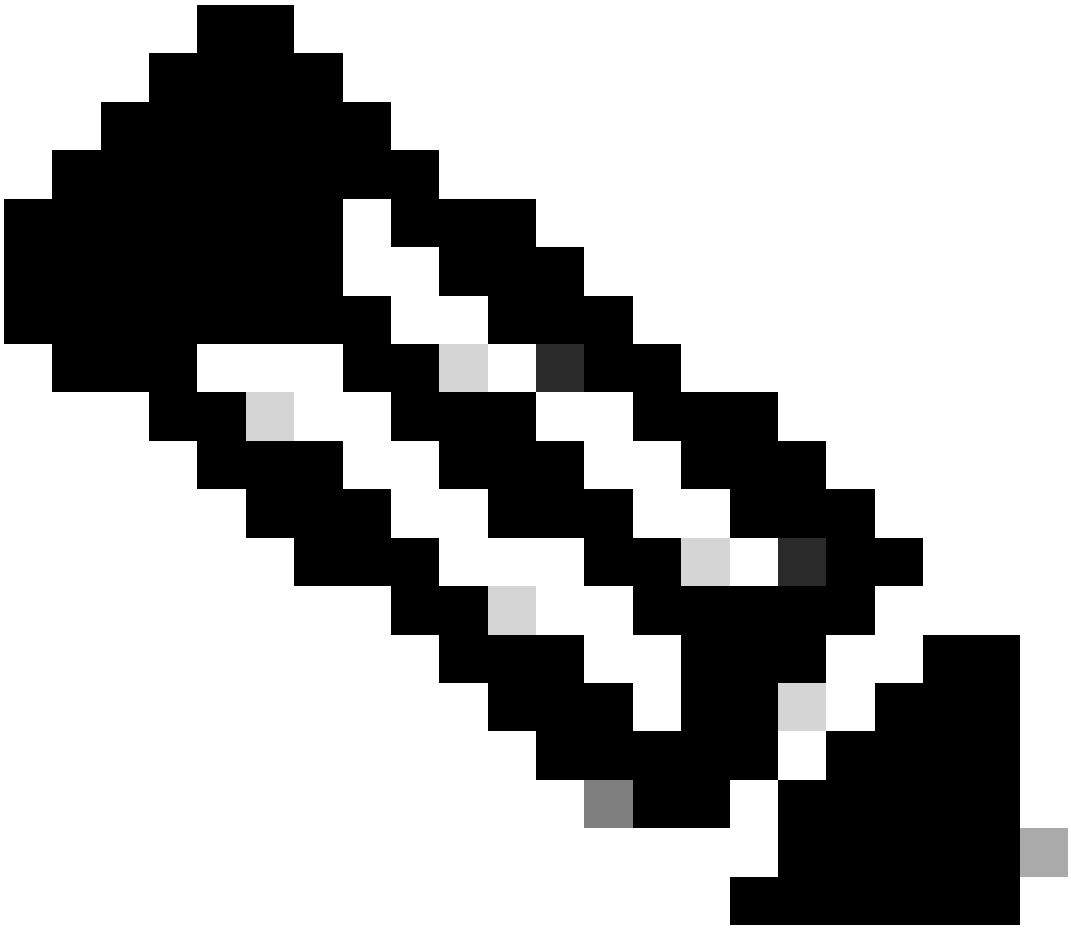
19. これで、コールをエージェントに転送できるようにするスクリプトのステップにコールが到達すると、CallSetupTransferRequestが表示されます。

20. 最初のメッセージとは異なり、これはリダイレクトではなく打診転送です。

21. これは打診転送である理由は、エージェントが受信可状態で席に着いていないときにコールをリダイレクトしても応答がないままになってしまうためです。打診転送を行うと、エージェントがいなければコールは再びキューに入ります。

22.

他のコンサルト転送と同様に、これはCTIポートが電話機の転送ボタンに初めて押し付けられたものです。



注: **ConsultWithoutMedia** は打診転送用のAPIです。

---

23. 通常のコンサルト転送では、AとCの間にメディアパスがありますが、UCCXとエージェントの間にメディアを確立する意味はないので、ここではCUCMにこれを行わないように指示します。

24. したがって、エージェントとCTIポート間でメディアカットオーバーを実行せずに打診転送を実行することになります。

25. この時点で、CTIポートは発信者を保留にし、コール状態が変更されたevent=HOLDを確認します。

26. ここで、CTIポートからエージェントデバイスへの新しいコールイベントが表示されます（元のコールIDを使用しますが、そ

のサブセットとP2イベントを使用します）。

27. P2イベントコールIDを検索すると、次のメッセージが「CallAnswer request」として表示されます。これは、エージェントがコールをピックアップしたことを意味します。

28. エージェントが（P2を使用して）コールをピックアップし、CTIポート側でもコールが接続状態になっている（P1を使用）ことがわかったら、ルートポイントにCallDirectTransferRequestが表示されます。これは、転送ボタンが2回目に押されたことを意味します。

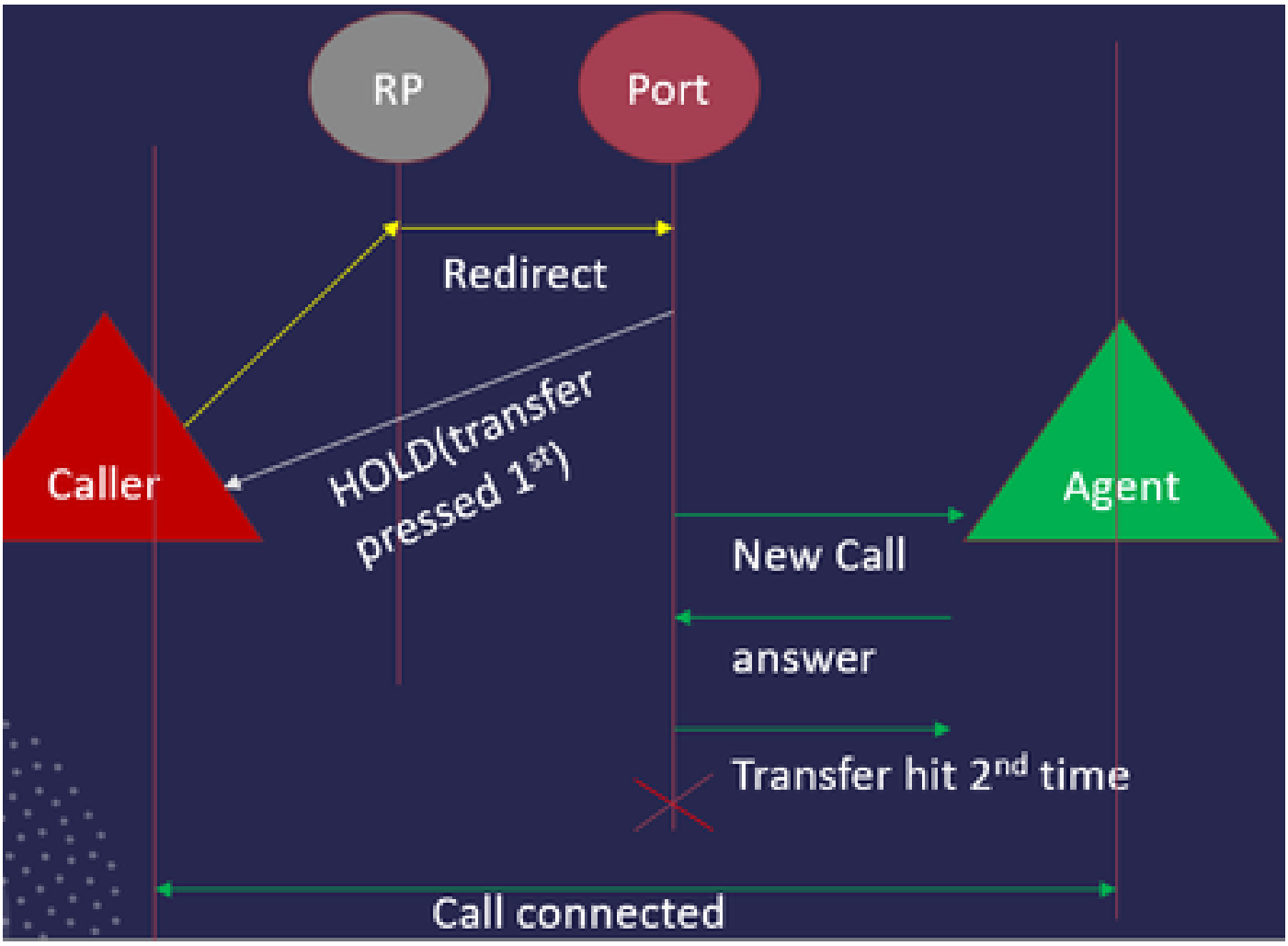
29. エージェントがコールをピックアップしたことをCTIポートが認識したため、ポイント待機が発生しなくなりました。このポートは、前述のように、2回目に転送ボタンを押すCTIポートであるCallDirectTransferRequest を直ちに送信します。

30. ここで、元のコールレッグ（保留中のコールレッグ）が残っています。

31. 他方のコールレッグが（ポートとエージェントの間で）破棄される。

32. この時点で、ゲートウェイを介して発信者とエージェントの間にCUCMとUCCXが認識されず、RTPが確立されます。

次の図は、前述のコールフローを要約したものです。



*JTAPI*コールフローの概要

トラブルシュート

JTAPIログの有効化と収集

JTAPIデバッグの有効化

JTAPIデバッグレベルを有効にするには、次の手順を確認してください。

- UCCXにログインします。
- **CCX Serviceability**に移動します。
- **Trace**に移動します。
- **Configuration**に移動します。
- Select Serviceドロップダウンから、**Cisco Unified CM Telephony Client**を選択します。
- **Debugging**チェックボックスを選択します。
- **MISC_DEBUGGING**以外のすべてのチェックボックスをオンにします。

JTAPIデバッグの収集

RTMTまたはCLIからJTAPIデバッグレベルを有効にするには、次の手順を確認してください。

RTMTから

- CCX Real Time Monitoring Toolにログインします。
- **Trace & Log Central**をクリックします。
- **Collect Files**をクリックします。
- すべてのサーバに対して**JTAPI Client**を選択します。
- [Next] をクリックします。
- ログファイルを保存するタイムスタンプと場所を選択します。

CLIから

JTAPIログの場所

activelog /uccx/log/JTAPI

JTAPIログを収集するコマンド：

file get activelog /uccx/log/JTAPI/*はcompressを繰り返します。

構文：

file get {activelog|inactivelog|install} file-spec [オプション]

転送するファイル仕様の必須ファイル

オプションのオプションreltime months|weeks|days|hours|minutes timevalue

abstime hh:mm:MM/DD/YY hh:mm:MM/DD/YY

正規表現の照合

再発

圧縮

タイムスタンプに基づいてログをダウンロードする5つの方法

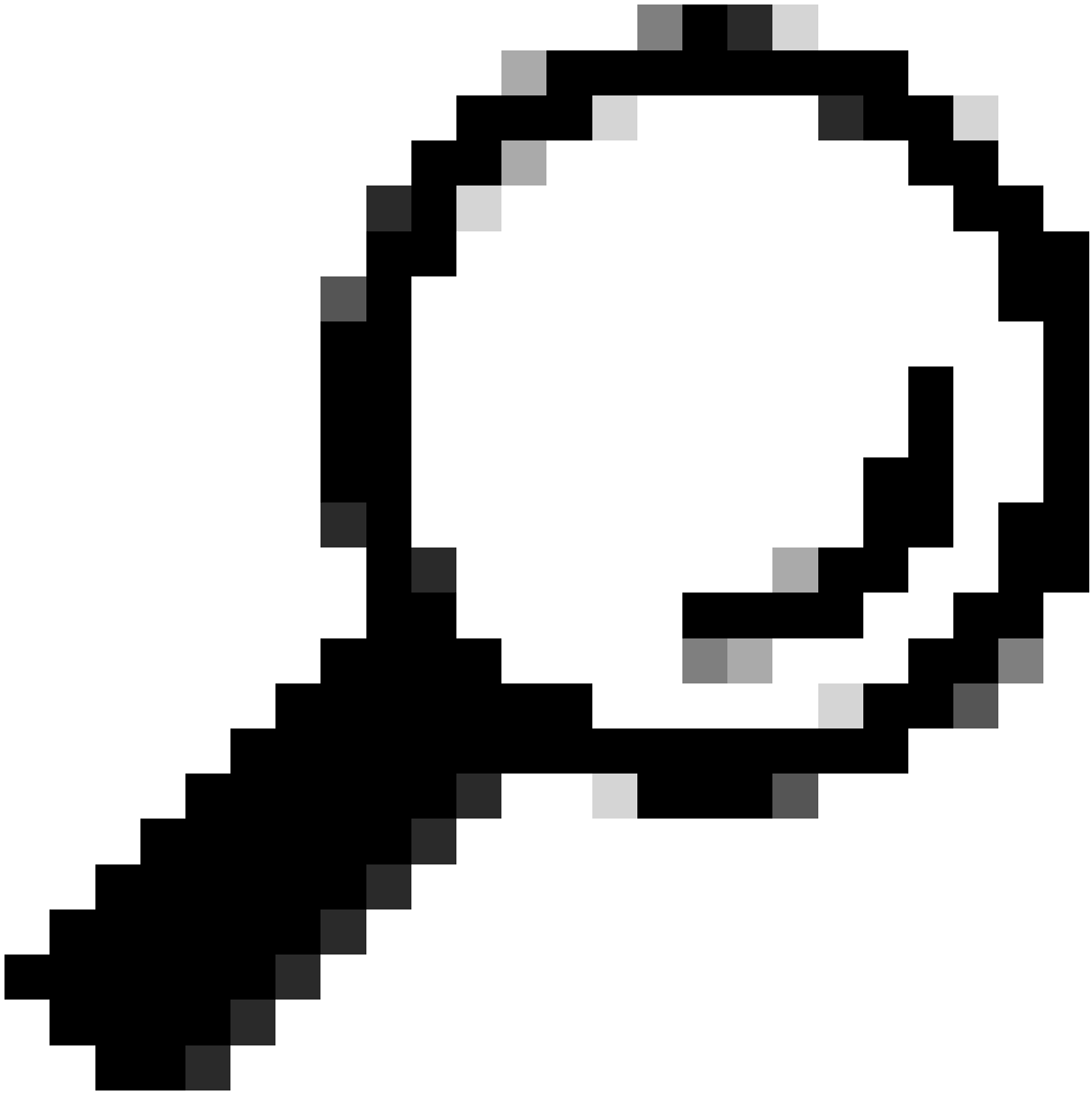
reltime：相対期間（分で指定）| 時間 | 日 | 数週間 | 月の値

abstime:hh:mm:MM/DD/YY hh:mm:MM/DD/YYで指定された絶対期間

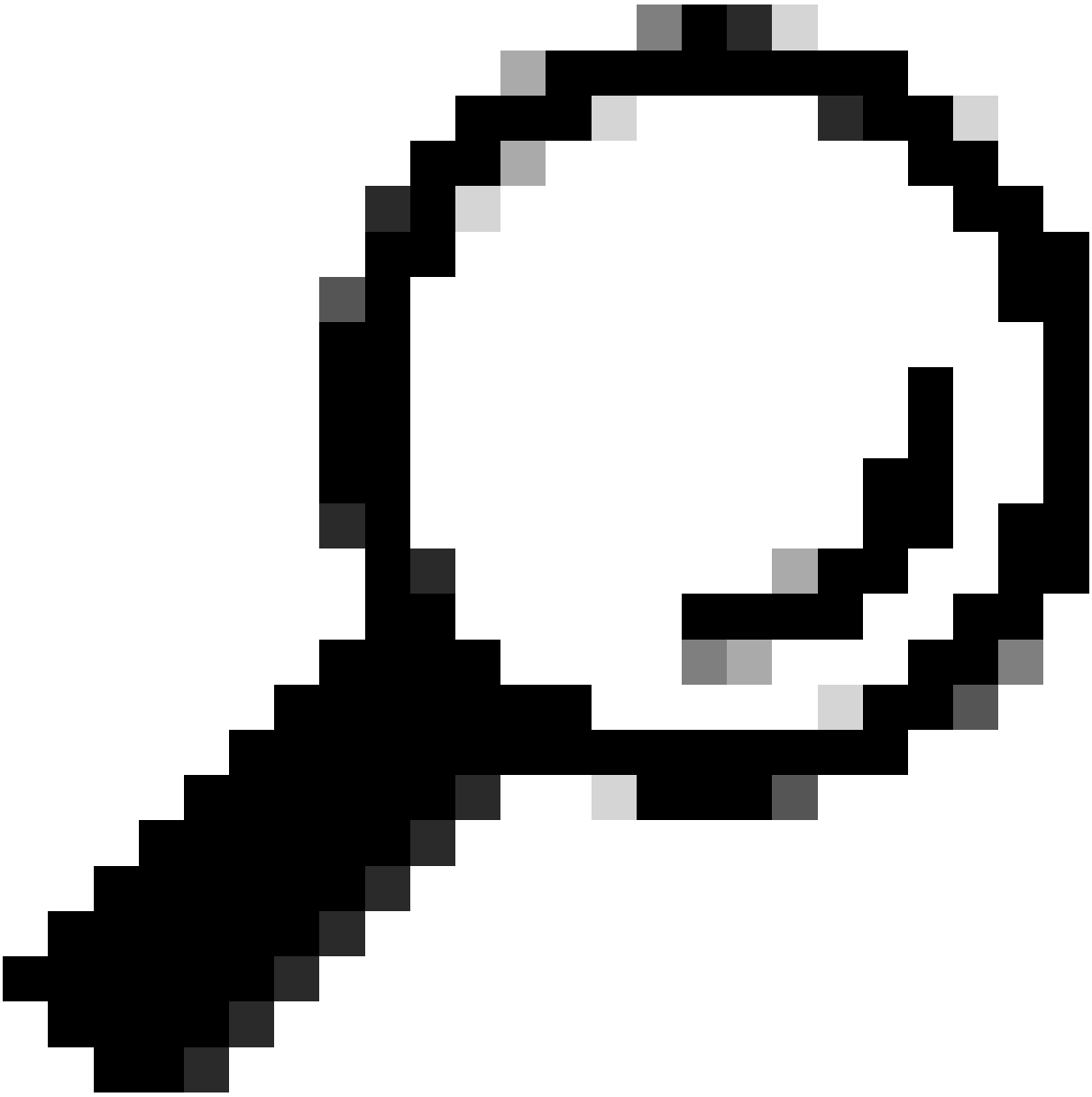
match：文字列値として指定されたファイル名の特定の文字列を照合します

recurs：サブディレクトリを含むすべてのファイルを取得する

compressオプションを使用すると、zip形式でファイルをダウンロードできます。



ヒント：ファイルをダウンロードするには、外部のSecure File Transfer Protocol(SFTP)サーバが設定されていて、アクセス可能であることを確認します。



ヒント: recursオプションを使用すると、ディレクトリ内のすべてのサブディレクトリとファイルを検索できます。これは、ディレクトリからすべてのログを取得する場合に使用します。

参照リンク

- JTAPI開発者ガイド

- UCCXトレースレベル

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。