



概要

Cisco Unified Communications Manager (Unified CM) は、Cisco Unified Communications ソリューションの高度なコール処理コンポーネントです。これは、スケーラブルで分散処理が可能な、しかも可用性の高いエンタープライズ IP テレフォニーコール処理ソリューションです。Unified CM はコミュニケーションのコラボレーションのプラットフォームの役割を果たしており、さまざまな機能をサポートしています。この高度なシステムのプロビジョニング、機能の呼び出し、モニタ、制御を行うため、Unified CM ではさまざまな種類のインターフェイスをサポートしています。

この章では、Unified CM の異なるインターフェイスの概要と、Cisco Unified Communications ソリューションの Java Telephony Application Programming Interface (JTAPI) アプリケーションを作成する前に理解しておく必要のある主な概念について、説明します。

Cisco Unified Communications Manager の機能については、[Cisco Unified JTAPI でサポートされる機能](#)を参照してください。詳細についてと、Computer–Telephony Integration (CTI) のデバイスおよびサポートされる機能については、[CTI でサポートされるデバイスとリリースごとの Cisco Unified JTAPI オペレーション](#)を参照してください。

- [Cisco Unified Communications Manager のインターフェイス \(1 ページ\)](#)
- [JTAPI の概要 \(5 ページ\)](#)
- [Cisco Unified JTAPI の概念 \(8 ページ\)](#)
- [コールバックのスレッド化 \(17 ページ\)](#)
- [アラーム サービス \(18 ページ\)](#)
- [ソフトウェア要件 \(19 ページ\)](#)
- [開発に関するガイドライン \(19 ページ\)](#)

Cisco Unified Communications Manager のインターフェイス

Unified CM でサポートされているインターフェイスは次の種類に分類できます。

- [プロビジョニングインターフェイス \(2 ページ\)](#)
- [デバイス監視およびコール制御インターフェイス \(3 ページ\)](#)

- サービスアビリティ インターフェイス (4 ページ)
- ルーティング規制インターフェイス (5 ページ)
- Cisco Unified Communications Manager のインターフェイス (1 ページ)

プロビジョニング インターフェイス

Unified CM のプロビジョニング インターフェイスには次のようなものがあります。

- Administrative XML
- Cisco エクステンション モビリティ サービス

Administrative XML

Administrative XML (AXL) API は、eXtensible Markup Language (XML) Simple Object Access Protocol (SOAP) インターフェイスを使用して、Unified CM 設定データベース内のデータの挿入、取得、更新、および削除を可能にするメカニズムです。AXL API を使用すると、プログラマはバイナリ ライブラリや DLL の代わりに XML を使用して Unified CM プロビジョニング サービスにアクセスし、データを XML 形式で交換できます。すべての AXL メソッド (要求) は HTTP と SOAP を組み合わせて実行します。SOAP は XML リモート プロシージャ コール プロトコルの一種です。ユーザは XML データを Unified CM パブリッシャ サーバに送信して要求を実行します。次に、パブリッシャから AXL 応答が返されます。この応答も SOAP メッセージです。詳細については、次の URL にある『Administrative XML Tech Center』を参照してください。<http://developer.cisco.com/web/axl/home>

Cisco エクステンション モビリティ

Cisco エクステンション モビリティ (Extension Mobility) サービスは Unified CM の機能で、この機能によりデバイス (通常は Cisco Unified IP Phone) は回線、短縮ダイヤル、サービスなどの新しいデバイスプロファイルを一時的に使用できます。ユーザは、他の Cisco Unified IP Phone からラインアピランス、サービス、短縮ダイヤルなどの各 Cisco Unified IP Phone 設定に一時的にアクセスできます。エクステンション モビリティ サービスは、新しい設定ファイルを電話機にダウンロードすることにより動作します。Unified CM はログイン ユーザについての情報に基づいて、この新しい設定ファイルを動的に生成します。アプリケーションで XML ベースのエクステンション モビリティ サービス API を使用すると、エクステンション モビリティ サービスの機能を利用することができます。

詳細は、Cisco Developer Network (<https://developer.cisco.com/site/extension-mobility/develop-and-test/documentation/latest-version/emapi-developer-guide.gsp>) の『Extension Mobility API Tech Center』を参照してください。

Unified CM の該当するリリースの『Cisco Unified Communications Manager XML Developers Guide』を参照してください。次のサイトで入手できます。

http://www.cisco.com/en/US/products/sw/voicesw/ps556/products_programming_reference_guides_list.html

デバイス監視およびコール制御インターフェイス

Unified CM のデバイス モニタリングおよび呼制御インターフェイスは次のとおりです。

- Cisco TAPI と Media Driver
- Cisco JTAPI
- Cisco Web Dialer

Cisco TAPI と Media Driver

Unified CM では Computer Telephony TAPI インターフェイス経由で IP テレフォニー デバイスと ソフトクライアントの高度な呼制御を行うことができます。シスコのテレフォニー サービス プロバイダー (TSP) と Media Driver インターフェイスにより、カスタム アプリケーションを使用してテレフォニー対応デバイスとコールイベントのモニタ、ファーストおよびサードパーティの呼制御の確立、メディアレイヤとの相互通信によるメディアの終了、音声案内の再生、コールの記録を行うことができます。

詳細は、Cisco Developer Network (<http://developer.cisco.com/web/axl/home>) の『TAPI and Wave Driver Tech Center』を参照してください。

Unified CM の該当するリリースの『Cisco Unified TAPI Developers Guide for Cisco Unified Communications Manager』を参照してください。次のサイトで入手できます。

http://www.cisco.com/en/US/products/sw/voicesw/ps556/products_programming_reference_guides_list.html

Cisco JTAPI

詳細については、Cisco Developer Network (<http://developer.cisco.com/web/jtapi/home>) および JTAPI の概要 (5 ページ) にある JTAPI Tech Center を参照してください。

Cisco Web Dialer

Web Dialer は Unified CM サーバにインストールされています。これを使用すると、Cisco Unified IP Phone ユーザは Web およびデスクトップ アプリケーションからコールを開始できます。たとえば、Web Dialer は社内電話帳にあるハイパーリンクされた電話番号を使用します。そのため、コールしようとしている相手の電話番号を Web ページでクリックすれば電話をかけることができます。Web Dialer の 2 つの主要コンポーネントは、Web Dialer Servlet と Redirector Servlet です。

詳細は、Cisco Developer Network (<https://developer.cisco.com/site/webdialer/develop-and-test/documentation/latest-version/>) の『Web Dialer Tech Center』を参照してください。

Cisco Web Dialer の詳細については、Unified CM の該当するリリースの『Cisco Unified Communications Manager XML Developers Guide』を参照してください。次のサイトで入手できます。

http://www.cisco.com/en/US/products/sw/voicesw/ps556/products_programming_reference_guides_list.html

サービスアビリティ インターフェイス

Unified CM のサービスアビリティ インターフェイスには次のようなものがあります。

- Serviceability XML
- SNMP/MIB

Serviceability XML

Unified CM Serviceability XML インターフェイス特有の問題のモニタ、診断、対処を行うためのサービスとツールの集合です。

- Unified CM ハードウェアおよびソフトウェアの健全性をモニタするための、プラットフォーム、サービス、アプリケーションの各パフォーマンス カウンタを備えている
- Unified CM に接続されている電話機、デバイス、アプリケーションの健全性をモニタするためにデバイスと CTI の接続ステータスをリアルタイムで表示する
- Unified CM サービスの遠隔操作（開始/停止/再開）を行うことができる
- トラブルシューティングと分析で使用する Unified CM トレース ファイルとログを収集してパッケージ化する
- 検索基準にもとづいてアプリケーションに呼詳細レコード ファイルを提供する
- 管理コンソールに Unified CM ハードウェアおよびソフトウェア特有の SNMP データを提供する

詳細は、Cisco Developer Network (<http://developer.cisco.com/web/sxml/home>) の『Serviceability XML Tech Center』を参照してください。

SNMP/MIB

SNMP インターフェイスを使用すると、外部アプリケーションによりさまざまな UCMgr エンティティに対するクエリーとレポート作成を実行できます。このインターフェイスの使用により、syslog 情報などの Unified Communication Manager の接続に関する情報がネットワーク内の他のデバイスに提供されます。

Unified CM でサポートされている MIB は次のとおりです。

- Cisco-CCM-MIB、CISCO-CDP-MIB、Cisco-syslog-MIB
- 標準の MIB（MIB II、SYSAPPL-MIB、HOST RESOURCES-MIB など）
- ベンダーの MIB

詳細は、Cisco Developer Network (<https://developer.cisco.com/site/sxml/>) の『SNMP/MIB Tech Center』を参照してください。

Unified CM の該当するリリースの『Cisco Unified Communications Manager XML Developers Guide』を参照してください。次のサイトで入手できます。

http://www.cisco.com/en/US/products/sw/voicesw/ps556/products_programming_reference_guides_list.html

ルーティング規制インターフェイス

Cisco Unified Communication Manager 8.0(1)以降は、外部コール制御（ECC）機能をサポートしています。この機能では、補助ルートサーバが、Cisco Unified Routing Rules Interface を使用して、Cisco Unified Communications Manager の代わりにコールルーティングを決定します。外部コール制御の設定に際して、Cisco Unified Communications Manager は、発信側および着信側の情報が入ったルート要求を補助ルートサーバに発行します。補助ルートサーバは、要求を受信し、適切なビジネスロジックを適用した後、Cisco Unified Communications Manager に対し、コールのルーティングとコールに適用すべきその他の取り扱い方法を指示するルート応答を返します。

詳細は、Cisco Developer Network (<https://developer.cisco.com/site/curri/develop-and-test/documentation/latest-version/>) の『Routing Rules Interface Tech Center』を参照してください。

Cisco Connection インターフェイス

このインターフェイスには、Connection オブジェクトで呼び出すことができる API があります。Connection では、コールとアドレスへの恒久的な参照が保持されます。コールイベントから取得した Connection の参照は、Connection のコール (`getCall ()`) とアドレス (`getAddress ()`) の取得に使用できます。

Cisco Unified Communications Manager の Cisco Connection インターフェイスには次のようなものがあります。

- Connection に関連付けられたパーティのローカルユニバーサルユニーク ID (UUID)
- コールの相手側に関連付けられたパーティのローカルユニバーサルユニーク ID (UUID)

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とともに使用する目的で、Sun Microsystems によって開発された標準のプログラミングインターフェイスとして機能します。Cisco JTAPI では、Sun JTAPI 1.2 仕様が Cisco の追加の機能拡張として実装されています。Cisco JTAPI を使用して、次のアプリケーションを開発できます。

- Cisco Unified Communications Manager 対応の電話機の監視と制御。
- Computer-Telephony Integration (CTI) ポートとルートポイント（仮想デバイス）を使用したコールのルーティング。

サポートされる基本的なテレフォニー API は、会議、転送、Connection、応答、リダイレクト API で構成されます。

`javax.telephony.*` 階層にある JTAPI インターフェイスのパッケージは、テレフォニーリソースを Java アプリケーションで操作するためのプログラミングモデルを定義しています。インター

フェイスの詳細については、[Cisco Unified JTAPI クラスとインターフェイス](#)を参照してください。

この章では、次のトピックについて取り上げます。

- [Cisco Unified JTAPI とコンタクトセンター](#) (6 ページ)
- [Cisco Unified JTAPI とエンタープライズ](#) (6 ページ)
- [Cisco Unified JTAPI アプリケーション](#) (7 ページ)
- [Jtprefs アプリケーション](#) (8 ページ)

Cisco Unified JTAPI とコンタクトセンター

Cisco Unified JTAPI は、コンタクトセンターで使用し、正しい時間に正しい場所へコールを送信するためにデバイスステータスのモニタとルーティング指示の発行を行い、分析用にコールの統計を取得すると同時に録音指示の停止や開始を指示し、さらに、CRMアプリケーション、自動化スクリプト、およびリモートコール制御内で、コールを画面にポップアップさせます。

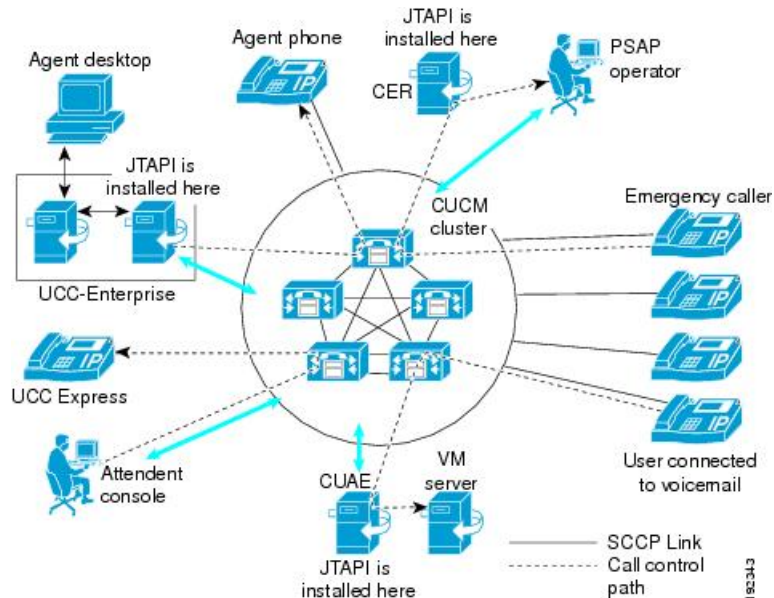
Cisco Unified JTAPI とエンタープライズ

Cisco Unified JTAPI はエンタープライズ環境で使用され、ユーザの応答可能、場所、プリファレンスを結合し、プレゼンスベースのルーティングを行うための固有に調整された環境を実現します。たとえば、金融環境では、ブローカーや分析者が国際金融市場の急激な変化に対応できるように、市場データ、ビジネスロジック、コール制御がブラウザベースのアプリケーションに組み込まれています。

健康管理環境では、コール制御、医者と患者の検索、救急隊の呼び出しが、ブラウザベースのコンソールに組み込まれています。さらに、サービス業環境では、発信者のデータが POS システムにリンクされ、部屋の予約またはレストランの予約、タクシーの手配、およびモーニングコールのスケジューリングなどの自動化が行われます。

次の図は、エンタープライズ用に設定された典型的な Cisco Unified Communications Manager と Cisco Unified JTAPI を示しています。

図 1 : Cisco Unified Communications Manager と Cisco Unified JTAPI



Cisco Unified JTAPI アプリケーション

Cisco Unified JTAPI アプリケーションのフローは次のようになります。

- JTAPIPeerFactory から JTAPIPeer オブジェクト インスタンスを取得する。
- JTAPIPeer で getProvider() API を使用し、プロバイダーを取得する。
- プロバイダーから、アプリケーションが使用する端末とアドレスを取得する。
- 関連するオブジェクトの機能を判別する。
- アプリケーションがモニタおよび制御する必要があるオブジェクトのオブザーバを追加する。
- アプリケーションフローを開始する（例：コールの開始）。

次の例に、基本的な JTAPI アプリケーションを示します。

```
public void getProvider () { try { JtapiPeer peer = JtapiPeerFactory.getJtapiPeer
( null ); System.out.println ("Got peer "+peer); Provider provider =
peer.getProvider ("cti-server;login = username;passwd = pass"); System.out.println
("Got provider "+provider); MyProviderObserver providerObserver = new
MyProviderObserver (); provider.addObserver(providerObserver); while
(outOfService ) { Thread.sleep(500); } System.out.println ("Provider is now in
service"); Address[] addresses = provider.getAddresses(); System.out.println
("Found "+ addresses.length +" addresses"); for(int i = 0; i< addresses.length;
i++) { System.out.println(addresses[i]); } provider.shutdown(); catch (Exception
e) { } }
```

Jtprefs アプリケーション

Cisco Unified JTAPI の設定に必要なパラメータは、`jtapi.ini` ファイルにあります。Cisco Unified JTAPI は、このファイルを Java クラスパス内で検出します。パラメータは、Cisco Unified JTAPI でインストールされる Jtprefs アプリケーションを使用して変更できます。Jtprefs アプリケーションではこのアプリケーションが必要とするパラメータだけが設定されます。`jtapi.ini` に依存せずに、1 箇所からアプリケーションを管理できるため、大変有用です。

`jtapi.ini` ファイルにはデフォルト値が含まれていますが、クライアントアプリケーションでは `jtapi.ini` ファイルを特に修正することなくさまざまな値を修正できます。ただし、クライアントアプリケーションのさまざまなインスタンスでは、これらのパラメータに対して個別の設定を指定できます。CiscoJtapiProperties インターフェイスは、`com.cisco.jtapi.extensions` パッケージで定義されています。

アプリケーションでは、CiscoJtapiPeer から CiscoJtapiProperties オブジェクトを取得し、アクセス用メソッドと変更用メソッドを使用してパラメータに変更を加えます。これらのプロパティは、CiscoJtapiPeer から派生したすべてのプロバイダーに対して、CiscoJtapiPeer で最初の `getProvider()` が呼び出される前に、設定および適用する必要があります。

`jtprefs.ini` を起動できない非 GUI ベースのプラットフォームで実行されるアプリケーションでは、`jtapi.ini` ファイルを作成して `jtapi.jar` とともに配置できます。

詳細については、次のトピックを参照してください。

- [JTAPI アプリケーションのユーザ情報の管理](#)
- [jtapi.ini ファイルのフィールド](#)

Cisco Unified JTAPI の概念

ここでは、次の概念について説明します。

- [CiscoObjectContainer インターフェイス \(9 ページ\)](#)
- [JtapiPeer と Provider \(9 ページ\)](#)
- [アドレスと端末の関係 \(11 ページ\)](#)
- [Connections \(12 ページ\)](#)
- [Terminal Connection \(13 ページ\)](#)
- [端末とアドレスの制限 \(13 ページ\)](#)
- [CiscoConnectionID \(17 ページ\)](#)

CiscoObjectContainer インターフェイス

CiscoObjectContainer インターフェイスを使用すると、このインターフェイスの実装されたオブジェクトに、アプリケーションによって定義されたオブジェクトを関連付けることができます。Cisco Unified JTAPI では、次のインターフェイス上で CiscoObjectContainer インターフェイスが拡張されます。

- CiscoJTAPIPeer
- CiscoProvider
- CiscoCall
- CiscoAddress
- CiscoTerminal
- CiscoConnection
- CiscoTerminalConnection
- CiscoConnectionID
- CiscoCallID

JtapiPeer と Provider

JtapiPeer オブジェクトの実装によって作成される Provider オブジェクトは、アプリケーションと JTAPI 実装間の主な接点として機能します。Provider オブジェクトは、アプリケーションで常に制御可能な Address、Terminal、Call などのコール モデル オブジェクト全体の集合を格納しています。

JTAPI Preferences (JTPREFS) アプリケーションでは、サーバ名を返す JtapiPeer.getServices() を管理します。

Provider は 2 つの基本プロセス（初期化とシャットダウン）を伴います。

アプリケーションで CiscoProvider を取得するときには、JtapiPeer.getProvider() メソッドを使用して必ず次の情報を渡します。

- Cisco Unified Communications Manager サーバのホスト名または IP アドレス
- ディレクトリで管理されるユーザのログイン
- 指定したユーザのパスワード
- (任意) アプリケーション情報 (このパラメータには任意の長さの文字列を指定できます)

appinfo が記録されてアラームが発生した場合に管理者がアラームの原因となったアプリケーションを認識できるように、アプリケーションには十分な情報を記述する必要があります。アプリケーションには、アプリケーションが常駐するホスト名または IP アドレス、およびアプリケーションが起動した時間を含めないでください。また、「=」および「;」は、getProvider() 文字列の区切りに使用されるため、いずれも appinfo 文字列に入れること

はできません。appinfo を指定しない場合、代わりに一般的な疑似固有名 (JTAPI[XXXX]@hostname) を使用できます (XXXX は 4 桁の乱数を表します)。

パラメータは、次のように文字列として連結されたキー値のペアで渡します。

```
JtapiPeer.getProvider("CTIManagerHostname;login = user;passwd = userpassword;appinfo = Cisco Softphone")
```

初期化

JtapiPeer.getProvider() メソッドでは、TCP リンク、Cisco Unified Communications Manager との初期ハンドシェイク、およびデバイスリストの列挙が完了すると同時に Provider オブジェクトが返されます。このとき、プロバイダーは OUT_OF_SERVICE 状態にあります。Cisco Unified JTAPI アプリケーションは、制御対象のデバイスリストが有効になる前に、プロバイダーが IN_SERVICE 状態に移行するのを待つ必要があります。ProvInServiceEv イベントは、ProviderObserver インターフェイスの実装されたオブジェクトに通知されます。



- (注) CiscoProviderObserver の実装だけでは不十分であり、provider.addObserver() を使用してプロバイダーにオブザーバを追加する必要があります。アプリケーションは、Provider がインサービス状態にあることを示す通知を待つ必要があります。

JTAPI における QoS ベースライン化作業の一環として、ProviderOpenCompletedEv は「アプリケーションの DSCP 値」を JTAPI に提供します。JTAPI は CTI との接続に対してこの DSCP 値を設定し、Provider オブジェクトが存在する限り、CTI へのすべての JTAPI メッセージがこの DSCP 値を保持します。

Shutdown

アプリケーションを使用して provider.shutdown() を呼び出すと、JTAPI では Cisco Unified Communications Manager との通信が恒久的に失われ、ProvShutdownEv イベントがアプリケーションに通知されます。Provider が再度起動されることはない想定し、アプリケーションによってシャットダウンを完全に処理する必要があります。

Provider.getTerminals()

このメソッドは、ディレクトリ上にあるユーザコントロールリストで管理されるデバイスに対して作成される端末の配列を返します。ユーザコントロールリストを管理するには、『Cisco Unified Communications Manager Administration Guide』を参照してください。

Provider.getAddresses()

このメソッドは、ディレクトリ上にあるユーザコントロールリストで管理されるデバイスに対して割り当てられる回線から作成されるアドレスの配列を返します。

ディレクトリ上のユーザコントロールリストの変更

JTAPIアプリケーションの起動後にユーザコントロールリストにデバイスが追加された場合、CiscoTermCreatedEv と各 CiscoAddrCreatedEv が生成され、CiscoProviderObserver の実装されたオブザーバに送信されます。また、アプリケーション側から、制御対象デバイスの現在の登録状態をモニタし、これらのデバイスのアベイラビリティを動的に追跡することができます。インサービスの Address または Terminal に対するイベントは、CiscoAddressObserver と CiscoTerminalObserver を実装しているオブザーバに通知されます。



(注) オブザーバの実装だけでは不十分であり、アドレスの場合は `address.addObserver()` メソッドを使用して、端末の場合は `terminal.addObserver()` メソッドを使用してオブザーバをそれぞれ追加する必要があります。



(注) `call.connect()` メソッドを呼び出す前に、発呼側のアドレスまたは端末に `CallObserver` を追加します。追加しないと、このメソッドによって例外が返されます。

アドレスと端末の関係

Cisco Unified Communications システムのアーキテクチャには3種類の基本エンドポイントがあります。

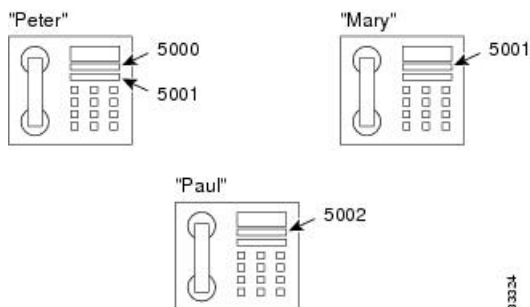
- 電話機
- 仮想デバイス（メディア終端点とルートポイント）
- ゲートウェイ

これらのエンドポイントのうち、電話機とメディア終端点だけが Cisco Unified JTAPI を実装することにより使用されます。

Cisco Unified Communications Manager を使用すると、ユーザは、1つ以上の回線やダイヤル可能な番号を電話機に設定して、その回線や番号を複数の電話機間で同時に共有したり、複数の回線を一度に1台の電話機だけで排他的に利用するように設定することができます。電話機には、回線あたり同時に複数のコールを終端させる機能 [（コール最大数（Maximum Number of Calls））の設定に依存] があり、1つのコール以外は保留になります。

これは、家庭用電話機の「「コール待機」」機能の動作に似ています。[図 2: 電話機の図 \(12 ページ\)](#) は、Peter と Mary が1本の電話回線 5001 を共有する設定と、Paul が専用の電話回線 5002 を使用する設定を示したものです。

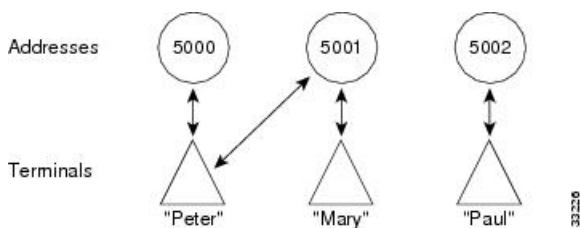
図 2: 電話機の図



すべてのタイプの Cisco Unified Communications Manager 端末が、一意の名前で識別されます。電話機の MAC アドレス（「SEP0010EB1014」など）で識別され、システム管理者は、一意の名前であれば、どのような名前でもメディアターミネーションポイントに割り当てることができます。

Cisco Unified JTAPI 実装では、プロバイダーによって制御される各エンドポイントごとに、管理者が割り当てた名前を使用して、対応する端末オブジェクトが構築されます。端末オブジェクトにもまた 1 つ以上のアドレスオブジェクトがあり、各アドレスオブジェクトはエンドポイント上の回線に対応します。図 1-2 の「アドレスと端末の関係」にアドレスと端末の関係を図示します。

図 3: アドレスと端末の関係



2 つ以上のエンドポイントで 1 本の回線（電話番号）を共有する場合は、対応するアドレスオブジェクトは 2 つ以上の端末オブジェクトに関連付けられます。

監視対象外の Address と Terminal

Cisco Unified JTAPI では、プロバイダーの端末とアドレスに CallObserver が付けられている場合にだけ、コールが認識されます。これは、Provider.getCalls() または Address.getConnections() などのメソッドでは、アドレスにコールがあってもこのアドレスに CallObserver が付けられていない場合は null が返されることを意味します。また、Call.connect() メソッドを使用して発呼するアドレスや端末にも、CallObserver を追加する必要があります。

Connections

Connection では、コールとアドレスへの恒久的な参照が保持されます。このため、コールイベントから取得した接続の参照は、常に接続のコール (getCall()) とアドレス (getAddress()) の取得に使用できます。

Terminal Connection

Terminal Connection では、常に端末と Connection への参照が保持されます。このため、コールイベントから取得した Terminal Connection の参照は、常に TerminalConnection (getTerminal()) と Connection (getConnection()) の取得に使用できます。

端末とアドレスの制限

この端末とアドレスの制限では、管理者が Cisco Unified Communications Manager Administration の制限リストに特定の端末とアドレスのセットを追加した場合、それらの端末とアドレスをアプリケーションで制御またはモニタすることが禁止されます。

管理者はデバイス上の特定の回線（特定の端末上のアドレス）を制限リストに追加できます。Cisco Unified Communications Manager Administration の制限リストに端末を追加した場合は、JTAPI でもその端末のすべてのアドレスが制限付きとしてマークされます。この設定が完了した後にアプリケーションを起動した場合は、CiscoTerminal.isRestricted() インターフェイスと CiscoAddress.isRestricted(Terminal) インターフェイスを確認することで、特定の端末またはアドレスが制限されているかどうかを認識することができます。共用回線の場合は、CiscoAddress.getRestrictedAddrTerminals() インターフェイスを照会することで、特定のアドレスがいずれかの端末で制限されているかどうかを確認できます。

アプリケーションの起動後に回線（端末上のアドレス）が制限リストに追加された場合は、CiscoAddrRestrictedEv がアプリケーションに通知されます。そのアドレスにオブザーバがある場合は CiscoAddrOutOfService がアプリケーションに通知されます。制限リストから回線が削除されると、CiscoAddrActivatedEv がアプリケーションに通知されます。そのアドレスにオブザーバがある場合は CiscoAddrInServiceEv がアプリケーションに送信されます。アプリケーションが制限リストに含まれるアドレスにオブザーバを追加しようとすると、PlatformException がスローされます。アドレスが制限される前に追加されたオブザーバはそのまま残りますが、そのアドレスが制限リストから削除されない限り、これらのオブザーバでイベントを取得できません。アプリケーションからアドレスのオブザーバを削除することもできます。

アプリケーションの起動後にデバイス（端末）が制限リストに追加された場合は、CiscoTermRestrictedEv がアプリケーションに送信されます。その端末にオブザーバがある場合は CiscoTermOutOfService がアプリケーションに送信されます。端末が制限リストに追加されると、JTAPI でもその端末に属するすべてのアドレスが制限され、CiscoAddrRestrictedEv がアプリケーションに通知されます。制限リストから端末が削除されると、CiscoTermActivatedEv と、対応するアドレスの CiscoAddrActivatedEv がアプリケーションに通知されます。アプリケーションが制限リストに含まれる端末にオブザーバを追加しようとすると、PlatformException がスローされます。端末が制限される前に追加されたオブザーバはそのまま残りますが、その端末が制限リストから削除されない限り、これらのオブザーバでイベントを取得できません。

アプリケーションの起動後に共用回線が制限リストに追加された場合は、CiscoAddrRestrictedOnTerminalEv がアプリケーションに通知されます。そのアドレスにアドレスオブザーバがある場合は、その端末の CiscoAddrOutOfServiceEv がアプリケーションに通知されます。すべての共用回線が制限リストに追加された場合は、最後の1つが追加された時点で、CiscoAddrRestrictedEv がアプリケーションに通知されます。アプリケーションの起動後に制限リストから共用回線が削除された場合は、CiscoAddrActivatedOnTerminalEv がアプリケーションに通知されます。そのアドレスにオブザーバがある場合は、その端末の

CiscoAddrInServiceEv がアプリケーションに通知されます。コントロールリストに含まれるすべての共用回線が制限リストから削除された場合は、最後の 1 つが削除された時点で CiscoAddrActivatedEv がアプリケーションに通知され、端末上のすべてのアドレスが InService イベントを受信します。

コントロールリストに含まれるすべての共用回線が制限付きとしてマークされている場合、アプリケーションがオブザーバを追加しようとすると、PlatformException がスローされます。一部の共用回線だけが制限リストに追加されている場合、アプリケーションがそのアドレスにオブザーバを追加すると、制限されていない回線だけがインサービスになります。

アドレスまたは端末が制限リストに追加されてリセットされたときにアクティブコールが存在していた場合は、Connection と TerminalConnections が接続解除として通知されます。

アドレスと端末が制限リストに 1 つも追加されていない場合、この機能は JTAPI の以前のバージョンと下位互換性をそのまま維持し、新しいイベントはアプリケーションに配信されません。

次のセクションでは、アドレスおよび端末の制限に関するインターフェイスの変更点について説明します。

CiscoTerminal

ブール値	<pre>isRestricted()</pre> <p>端末が制限されているかどうかを示します。端末が制限されている場合、その端末に関連付けられたすべてのアドレスも制限されます。端末が制限されている場合は true を返します。端末が制限されていない場合は false を返します。</p>
------	--

CiscoAddress

javax.telephony.Terminal[]	<pre>getRestrictedAddrTerminals()</pre> <p>このアドレスが制限されている端末の配列を返します。制限されていない端末がない場合、このメソッドは null を返します。</p> <p>共用回線の場合、端末上の少数の回線が制限されている可能性があります。このメソッドは、このアドレスが制限されているすべての端末を返します。アプリケーションは、制限された回線のコールイベントを認識できません。制限された回線が他の制御デバイスとのコールに関わっている場合、制限された回線用の外部 Connection が作成されます。</p>
ブール値	<pre>isRestricted(javax.telephony.Terminal terminal)</pre> <p>この端末上のいずれかのアドレスが制限されている場合、true を返します。この端末上のアドレスが制限されていない場合、false を返します。</p>

```
public interface CiscoRestrictedEv extends CiscoProvEv {
    public static final int ID = com.cisco.jtapi.CiscoEventID.CiscoRestrictedEv;

    /**
```

```

* The following define the cause codes for restricted events
*/

public final static int CAUSE_USER_RESTRICTED = 1;

public final static int CAUSE_UNSUPPORTED_PROTOCOL = 2;

}

```

これは制限されたイベントの基底クラスを表し、すべての制限されたイベントの原因コードを定義します。CAUSE_USER_RESTRICTEDは、端末またはアドレスが制限とマークされていることを示します。CAUSE_UNSUPPORTED_PROTOCOLは、コントロールリスト内のデバイスがCisco Unified JTAPIでサポートされていないプロトコルを使用していることを示します。SIPを実行している既存のCisco Unified IP 7960 フォンおよび7940 フォンはこれに該当します。

CiscoAddrRestrictedEv

public interface CiscoAddrRestrictedEv extends CiscoRestrictedEv. アプリケーションは、回線または関連するデバイスがCisco Unified Communications Manager Administrationから制限されたときに、このイベントを通知します。制限された回線では、アドレスがアウトオブサービスになり、再び有効になるまでインサービスに戻りません。アドレスが制限されている場合は、addCallObserver および addObserver によって例外がスローされます。共用回線では、いくつかの回線だけが制限されて残りは制限されなかった場合、例外はスローされませんが、制限された共用回線はイベントを受け取りません。すべての共用回線が制限された場合は、オブザーバを追加すると例外がスローされます。オブザーバを追加した後にアドレスが制限された場合、アプリケーションはCiscoAddrOutOfServiceEvを認識し、アドレスが有効にされるとインサービスになります。

CiscoAddrActivatedEv

public interface CiscoAddrActivatedEv extends CiscoProvEv. アプリケーションは、回線または関連するデバイスがコントロールリストに含まれており、Cisco Unified Communications Manager Administrationの制限リストから削除されたときに、このイベントを識別します。該当アドレスにオブザーバが存在する場合、アプリケーションはCiscoAddrInServiceEvを識別します。オブザーバが存在しない場合、アプリケーションはオブザーバの追加を試みるのが可能で、アドレスはインサービス状態になります。

CiscoAddrRestrictedOnTerminalEv

public interface CiscoAddrRestrictedOnTerminalEv extends CiscoRestrictedEv. ユーザがコントロールリストに共有アドレスを持っており、いずれかの回線を制限リストに追加した場合、このイベントが送信されます。getTerminal() インターフェイスは、アドレスが制限される端末を返します。getAddress() インターフェイスは、制限されるアドレスを返します。

javax.telephony.Address	getAddress()
javax.telephony.Terminal	getTerminal()

CiscoAddrActivatedOnTerminal

public interface **CiscoAddrActivatedOnTerminalEv** extends CiscoProvEv。共有回線または共有回線を持つデバイスを制限リストから削除すると、このイベントが送信されます。getTerminal() インターフェイスは、アドレスに追加される端末を返します。getAddress() インターフェイスは、新しい端末が追加されるアドレスを返します。

javax.telephony.Address	getAddress ()
javax.telephony.Terminal	getTerminal ()

CiscoTermRestrictedEv

public interface **CiscoTermRestrictedEv** extends CiscoRestrictedEv。アプリケーションは、アプリケーションの起動後にデバイスが Cisco Unified Communications Manager Administration から制限リストに追加されたときに、このイベントを認識します。アプリケーションは、制限された端末やその端末のアドレスに関するイベントを識別できません。InService 状態にある端末が制限された場合、アプリケーションはこのイベントを受信し、端末およびそれに対応するアドレスはアウト オブ サービス状態になります。

CiscoTermActivatedEv

public interface **CiscoTermActivatedEv** extends CiscoRestrictedEv。

javax.telephony.Terminal	getTerminal () 有効化されて制限リストから削除された端末を返します。
--------------------------	--

CiscoOutOfServiceEv

static int	CAUSE_DEVICE_RESTRICTED デバイスが制限されたためにイベントが送られたかどうかを示します。
static int	CAUSE_LINE_RESTRICTED 回線が制限されたためにイベントが送られたかどうかを示します。

CiscoCallEv

static int	CAUSE_DEVICE_RESTRICTED デバイスが制限されたためにイベントが送られたかどうかを示します。
static int	CAUSE_LINE_RESTRICTED 回線が制限されたためにイベントが送られたかどうかを示します。

CiscoConnectionID

CiscoConnectionID オブジェクトは、Cisco Unified JTAPI の各 Connection に関連付けられた固有のオブジェクトを表します。アプリケーションでは、オブジェクト自体またはオブジェクトの整数表現を使用できます。

コールバックのスレッド化

Cisco Unified JTAPI 実装の設計では、アプリケーションによって `Call.connect()` と `TerminalConnection.answer()` などのブロッキング用 JTAPI メソッドを、そのオブザーバのコールバック内から呼び出すことができます。これは、オブザーバのコールバック内から JTAPI メソッドを使用しないように警告する JTAPI 1.2 仕様の制限に、アプリケーションが制約されないことを意味します。

CiscoSynchronousObserver インターフェイス

Cisco Unified JTAPI 実装では、アプリケーションによって `Call.connect()` と `TerminalConnection.answer()` などのブロッキング用 JTAPI メソッドを、オブザーバのコールバック内から呼び出すことができます。これは、オブザーバのコールバック内部から JTAPI メソッドを使用しないように警告する JTAPI 1.2 仕様の制限に、アプリケーションが制約されないことを意味します。アプリケーションでは、そのオブザーバオブジェクト上に `CiscoSynchronousObserver` インターフェイスを実装して、Cisco Unified JTAPI 実装のキューイングロジックを選択的に無効にできます。

多くのアプリケーションは、この非同期動作の悪影響を受けることはありません。オブザーバコールバック中にコヒーレントコールモデルの機能を使用するアプリケーションでは、Cisco Unified JTAPI 実装のキューイングロジックを選択的に無効にできます。対象のオブザーバオブジェクト上に `CiscoSynchronousObserver` インターフェイスを実装することで、アプリケーションからそのオブザーバに通知同期イベントが宣言されます。同期オブザーバに通知されるイベントは、オブザーバコールバック内から照会されるコールモデルオブジェクトの状態と一致します。



- (注) `CiscoSynchronousObserver` インターフェイスの実装されたオブジェクトでは、そのイベントコールバック内部からブロッキング用 JTAPI メソッドを呼び出さないでください。これを行った場合の影響は予測不能であり、JTAPI 実装が停止する可能性もあります。一方、`Call.getState()` または `Connection.getState()` などのすべての JTAPI オブジェクトのアクセス用メソッドは安全に使用できます。アプリケーションでは、同期コールバックで `Terminal.getAddresses()` などの配列を返すインターフェイスを呼び出さないよう回避する必要があります。

動的オブジェクトの照会

コールオブジェクトのような動的オブジェクトの照会には注意が必要です。イベントを取得する時間までに、オブジェクト（コールなど）は示された状態とは別の状態で存在している場合があります。たとえば、CiscoTransferStartEVを取得する時間までに、転送コールによってその内部 Connection がすべて削除されている場合があります。

callChangeEvent()

callChangedEvent() メソッドが呼び出されるときには、イベントに格納された参照の妥当性は引き続き保証されます。たとえば、イベントに getConnection() メソッドがある場合、アプリケーションでこのメソッドを呼び出して有効な Connection の参照を取得できます。同様に、getCallingAddress() メソッドでは有効な Address オブジェクトを返すことが保証されます。

CiscoConsultCall

CiscoConsultCall インターフェイスの場合、コンサルティング TerminalConnection への参照が恒久的に保持されます。たとえば、CiscoConsultCallActive イベントの処理時では、getConsultingTerminalConnection() は有効な TerminalConnection の参照を返すことを保証します。また、TerminalConnection は、コンサルティング Connection とそのコンサルティング コールへのアクセスを保証します。

CiscoTransferStartEv

CiscoTransferStartEv の場合、callChangedEvent() が呼び出されると、イベント内にある転送コール、転送コントローラおよび最後のコールへの参照が有効になります。ただし、getConnections() を呼び出したときには、getConnections() から Connection が返される場合と返されない場合があります。

アラーム サービス

Cisco Unified Communications アプリケーションに対する一般的なサービスフレームワークの一部として、サービスへのアラームの送信がサポートされています。com.cisco.services.alarm パッケージでは、アラーム コンポーネントが定義されています。

アラームのインターフェイスとフレームワークでは、Cisco Unified JTAPI アプリケーションのネットワーク上で使用可能なアラーム サービスへ、Extensible Markup Language (XML) 形式のアラーム通知を TCP 経由で送信することがサポートされます。アラーム パッケージには、次の機能があります。

- アラーム サービスのカタログで解決される、アラームの XML 定義
- 送信側でアラームをバッファする、制限があり上書きされるキュー
- 送信アプリケーション側でブロッキングを回避する、別のスレッド上でのアラーム送信
- アラーム サービスへの TCP ベースの再接続スキーム

Cisco Unified JTAPI アラーム システムの全体的なフレームワークは、既存の JTAPI トレース パッケージに似た点が含まれています。アプリケーションでは、アラームオブジェクトを作成可能な特定のファシリティ コード用の AlarmManager をインスタンスにする必要があります。実装の一部として、DefaultAlarm および DefaultAlarmWriter 実装クラスが含まれています。

ソフトウェア要件

次の表では、JTAPIアプリケーション、JTPREFS、およびサンプルコードのソフトウェア要件を示しています。

アプリケーション	ソフトウェア要件
JTAPI アプリケーション	いずれかの JDK 1.4.2 対応 Java 環境
JTPREFS	いずれかの JDK 1.4.2 対応環境
コード例	いずれかの JDK 1.4.2 対応 Java 環境

開発に関するガイドライン

シスコは、Cisco Unified Communications Manager (Cisco Unified CM) の少なくとも 1 つ前の主要リリースに対してインターフェイスの下位互換性のポリシーを維持しています。さらに、シスコでは、Cisco Unified CM の新しい主要リリースへの互換性の維持を目的として、Cisco Solution Partner Program 加入者のアプリケーションでは再テストとアップデートを行うことを推奨します。★ICR指示により変更。FY14-0304。「Cisco Technology Developer Program」→「Cisco Solution Partner Program」★

必要なアップデートの件数と範囲を減らすために、すべての開発者（Cisco Solution Partner Programの開発者を含む）には、次の対策を推奨しています。★ICR指示により変更。FY14-0304。「Cisco Technology Developer Program」→「Cisco Solution Partner Program」★

- イベントやメッセージの順番は変わる場合があります。開発者は、イベントまたはメッセージの順番に依存しないでください。たとえば、機能の呼び出しのときに、独立したトランザクションが複数ある場合は、イベントまたはメッセージの前後関係が交差することがあります。2 番目のトランザクションに関連するイベントが、最初のトランザクションに関連するメッセージの前に発生することがあります。また、インターフェイスの制御が及ばないために（ネットワークや転送の障害など）、イベントまたはメッセージが遅延する場合があります。プロトコル オペレーションにおいて順序が必須の場合でも、アプリケーションが、順序が入れ替わったイベントやメッセージを処理できることが必要です。
- インターフェイスのイベントまたはメッセージ内の要素の順序は、プロトコル仕様の制約の範囲内で変わる場合があります。開発者は、必要でない限り、情報を解釈する目的で要素の順序に依存しないようにしてください。

- インターフェイスのイベント、メソッド、応答、ヘッダー、パラメータ、属性などの要素が新規に追加されたり、既存要素に新しい値が設定されたりすることがあります。開発者は、新規の要素または既存要素の新しい値に対し、必要に応じて無視するか、一般的な処置をとる必要があります。
- インターフェイスの既存のイベント、メソッド、応答、ヘッダー、パラメータ、属性などの要素はそのままにし、障害の修復が必要であれば、整合性をとって、可能な限り、元の意味と動作を保持します。
- 障害を修復した後はインターフェイスの動作が変わる場合があるため、障害に起因する動作（公開されたインターフェイス仕様と整合性がとれない動作）にアプリケーションが影響を受けないようにする必要があります。
- 推奨されなくなったメソッド、ハンドラ、イベント、応答、ヘッダー、パラメータ、属性は、それらが Cisco Unified CM から削除された場合に問題とならないように、できる限り迅速に、それらの使用をアプリケーションから削除する必要があります。
- アプリケーションの開発者は、新規機能および新しくサポートされるデバイス（電話機など）には上位互換性のないものがあることに注意する必要があります。新規機能および新規デバイスを使用する場合、互換性保持のためや、新規機能と新規デバイス利用のために、アプリケーションの変更が必要になることがあります。