



# CHAPTER 1

## Overview

---

Cisco Unified Communications Manager is the powerful call-processing component of the Cisco Unified Communications Solution. It is a scalable, distributable, and highly available enterprise IP telephony call-processing solution. Cisco Unified Communications Manager acts as the platform for collaborative communication and as such supports a wide array of features. In order to provision, invoke the features, monitor, and control such a powerful system, Cisco Unified Communications Manager supports different interface types.

This chapter also describes the major concepts of Cisco Unified TAPI service provider (Cisco Unified TSP) implementation. It contains the following sections:

- [Cisco Unified Communications Manager Interfaces, page 1-1](#)
- [Cisco Unified TSP Overview, page 1-4](#)
- [Cisco Unified TSP Concepts, page 1-5](#)
- [Development Guidelines, page 1-13](#)

## Cisco Unified Communications Manager Interfaces

The interface types supported by Cisco Unified Communications Manager are divided into the following types:

- [Provisioning Interfaces, page 1-1](#)
- [Device Monitoring and Call Control Interfaces, page 1-2](#)
- [Serviceability Interfaces, page 1-3](#)
- [Routing Rules Interface, page 1-4](#)

## Provisioning Interfaces

The following are the provisioning interfaces of Cisco Unified Communications Manager:

- Administration XML
- Cisco Extension Mobility service

## Administrative XML

The Administration XML (AXL) API provides a mechanism for inserting, retrieving, updating and removing data from the Cisco Unified Communications Manager configuration database using an eXtensible Markup Language (XML) Simple Object Access Protocol (SOAP) interface. This allows a programmer to access Unified CM provisioning services using XML and exchange data in XML form, instead of using a binary library or DLL. The AXL methods, referred to as requests, are performed using a combination of HTTP and SOAP. SOAP is an XML remote procedure call protocol. Users perform requests by sending XML data to the Cisco Unified Communications Manager Publisher server. The publisher then returns the AXL response, which is also a SOAP message.

For more information, See the Administrative XML Tech Center on the Cisco Developer Network <http://developer.cisco.com/web/axl/home>.

## Cisco Extension Mobility

The Cisco Extension Mobility (Extension Mobility) service, a feature of Cisco Unified Communications Manager, allows a device, usually a Cisco Unified IP Phone, to temporarily embody a new device profile, including lines, speed dials, and services. It enables users to temporarily access their individual Cisco Unified IP Phone configuration, such as their line appearances, services, and speed dials, from other Cisco Unified IP Phones. The Extension Mobility service works by downloading a new configuration file to the phone. Cisco Unified Communications Manager dynamically generates this new configuration file based on information about the user who is logging in. You can use the XML-based Extension Mobility service API with your applications, so they can take advantage of Extension Mobility service functionality.

For more information, See the Extension Mobility API Tech Center on the Cisco Developer Network <http://developer.cisco.com/web/emapi/home>.

Also, see *Cisco Unified Communications Manager XML Developers Guide* for relevant release of Cisco Unified Communications Manager at the following location:

[http://www.cisco.com/en/US/products/sw/voicesw/ps556/products\\_programming\\_reference\\_guides\\_list.html](http://www.cisco.com/en/US/products/sw/voicesw/ps556/products_programming_reference_guides_list.html).

## Device Monitoring and Call Control Interfaces

The following are the device monitoring and call control interfaces of Cisco Unified Communications Manager:

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

### Cisco TAPI and Media Driver

Cisco Unified Communications Manager exposes sophisticated call control of IP telephony devices and soft-clients using the Computer Telephony TAPI interface. Cisco's Telephone Service Provider (TSP) and Media Driver interface enables custom applications to monitor telephony-enabled devices and call events, establish first- and third-party call control, and interact with the media layer to terminate media, play announcements, record calls.

For more information, see the TAPI and Media Driver Tech Center on the Cisco Developer Network at the following location:

<http://developer.cisco.com/web/tapi/home>.

## Cisco JTAPI

For more information, see the JTAPI Tech Center on the Cisco Developer Network at the following location:

<http://developer.cisco.com/web/jtapi/home>.

Also, see *Cisco Unified JTAPI Developers Guide for Cisco Unified Communications Manager* for relevant release of Cisco Unified Communications Manager at the following location:

[http://www.cisco.com/en/US/products/sw/voicesw/ps556/products\\_programming\\_reference\\_guides\\_list.html](http://www.cisco.com/en/US/products/sw/voicesw/ps556/products_programming_reference_guides_list.html)

## Cisco Web Dialer

The Web Dialer, which is installed on a Cisco Unified Communications Manager server, allows Cisco Unified IP Phone users to make calls from web and desktop applications. For example, the Web Dialer uses hyperlinked telephone numbers in a company directory to allow users to make calls from a web page by clicking the telephone number of the person that they are trying to call. The two main components of Web Dialer comprise the Web Dialer Servlet and the Redirector Servlet.

For more information, see the Web Dialer Tech Center on the Cisco Developer Network

<http://developer.cisco.com/web/wd/home>.

For more information on Cisco Web Dialer, see *Cisco Unified Communications Manager XML Developers Guide* for relevant release of Cisco Unified Communications Manager at the following location:

[http://www.cisco.com/en/US/products/sw/voicesw/ps556/products\\_programming\\_reference\\_guides\\_list.html](http://www.cisco.com/en/US/products/sw/voicesw/ps556/products_programming_reference_guides_list.html).

## Serviceability Interfaces

The following are the serviceability interfaces of Cisco Unified Communications Manager:

- Serviceability XML
- SNMP/MIBs

### Serviceability XML

A collection of services and tools designed to monitor, diagnose, and address issues specific to Unified CM. serviceability XML interface:

- Provides platform, service and application performance counters to monitor the health of Unified CM hardware and software
- Provides real-time device and Computer Telephony Integration (CTI) connection status to monitor the health of phones, devices, and applications connected to Cisco Unified Communications Manager.
- Enables remote control (Start/Stop/Restart) of Cisco Unified Communications Manager services.

- Collects and packages Cisco Unified Communications Manager trace files and logs for troubleshooting and analysis.
- Provides applications with Call Detail Record files based on search criteria.
- Provides management consoles with SNMP data specific to Cisco Unified Communications Manager hardware and software.

For more information, see the Serviceability XML Tech Center on the Cisco Developer Network <http://developer.cisco.com/web/sxml/home>.

## SNMP/MIBs

SNMP interface allows external applications to query and report various UCMgr entities. It provides information on the connectivity of the Unified Communication Manager to other devices in the network, including syslog information.

The MIBs supported by Cisco Unified Communications Manager includes:

- Cisco-CCM-MIB, CISCO-CDP-MIB, Cisco-syslog-MIB
- Standard MIBs like MIB II, SYSAPPL-MIB, HOST RESOURCES-MIB
- Vendor MIBs

For more information, see the SNMP/MIB Tech Center on the Cisco Developer Network <http://developer.cisco.com/web/sxml/home>.

Also, see *Cisco Unified Communications Manager XML Developers Guide* for relevant release of Cisco Unified Communications Manager at the following location:

[http://www.cisco.com/en/US/products/sw/voicesw/ps556/products\\_programming\\_reference\\_guides\\_list.html](http://www.cisco.com/en/US/products/sw/voicesw/ps556/products_programming_reference_guides_list.html).

## Routing Rules Interface

Cisco Unified Communication Manager 8.0(1) and later supports the external call control (ECC) feature, which enables an adjunct route server to make call-routing decisions for Cisco Unified Communications Manager by using the Cisco Unified Routing Rules Interface. When you configure external call control, Cisco Unified Communications Manager issues a route request that contains the calling party and called party information to the adjunct route server. The adjunct route server receives the request, applies appropriate business logic, and returns a route response that instructs Cisco Unified Communications Manager on how the call should get routed, along with any additional call treatment that should get applied.

For more information, see the Routing Rules Interface Tech Center on the Cisco Developer Network <http://developer.cisco.com/web/curri/home>.

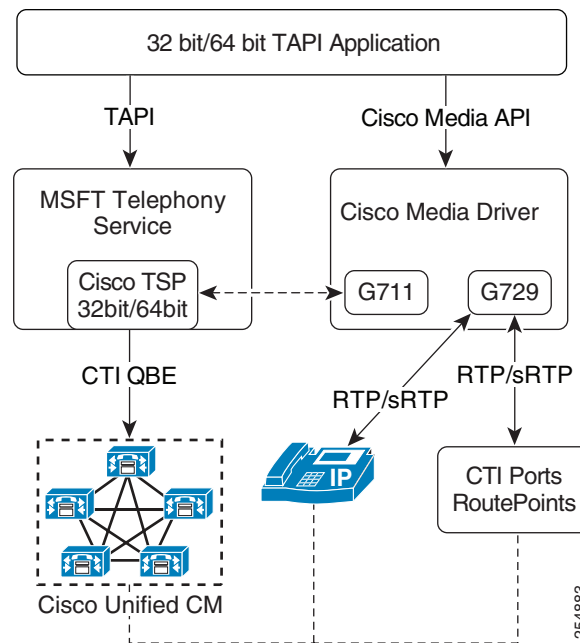
# Cisco Unified TSP Overview

The standard TAPI provides an unchanging programming interface for different implementations. The goal of Cisco in implementing TAPI for the Cisco Unified Communications Manager platform remains to conform as closely as possible to the TAPI specification, while providing extensions that enhance TAPI and expose the advanced features of Cisco Unified Communications Manager to applications.

As versions of Cisco Unified Communications Manager and Cisco Unified TSP are released, variances in the API should be minor and should tend in the direction of compliance. Cisco stays committed to maintaining its API extensions with the same stability and reliability, though additional extensions may be provided as new Cisco Unified Communications Manager features become available.

Figure 1-1 shows the architecture of TAPI.

**Figure 1-1 Architecture of TAPI Service Process**



**Note**

The Cisco TSP is a TAPI 2.1 service provider.

## Cisco Unified TSP Concepts

The following are described in this section:

- [Basic TAPI Applications](#), page 1-6
- [Cisco TSP Components](#), page 1-6
- [Cisco Media Drivers](#), page 1-7
- [TAPI Debugging](#), page 1-7
- [Cisco TSP Components](#), page 1-6

See “Basic TAPI Implementation” section on page 5-1 and “Cisco Device-Specific Extensions” section on page 6-1 for lists and descriptions of interfaces and extensions.

## Basic TAPI Applications

Microsoft has defined some basic APIs which can be invoked/supported from application code. All Microsoft defined APIs that can be used from the TAPI applications are declared in TAPI.H file. TAPI.H file is a standard library file that is with the VC++/VS2005 Installation. For example, C:\Program Files\Microsoft Visual Studio\VC98\Include\TAPI.H.

To use any specific API which is added or provided by Cisco TSP, the application needs to invoke that API by using the LineDevSpecific API.

### Simple Application

```
#include <tapi.h>
#include <string>
#include "StdAfx.h"
class TapiLineApp {
LINEINITIALIZEEXPARAMS mLineInitializeExParams;//was declared in TAPI.h files
    HLINEAPP    mhLineApp;
    DWORD      mdwNumDevs;
    DWORD      dwAPIVersion = 0x20005

public:
    // App Initialization
    // Note hInstance can be NULL
    // appstr - value can be given the app name "test program"
    bool TapiLineApp::LineInitializeEx(HINSTANCE hInstance, std::string appStr)
    {
        unsigned long lReturn = 0;
        mLineInitializeExParams.dwTotalSize = sizeof(mLineInitializeExParams);
        mLineInitializeExParams.dwOptions = LINEINITIALIZEEXOPTION_USEEVENT;
        lReturn = lineInitializeEx (&mhLineApp, hInstance, NULL, appStr.c_str),
        &mdwNumDevs,&dwAPIVersion,&LineInitializeExParams);
        if ( lReturn == 0 ) {
            return true;
        }
        else {
            return false;
        }
    }
    //App shutdown
    bool TapiLineApp::LineShutdown()
    {
        return! (lineShutdown (mhLineApp));
    }
};
```

## Cisco TSP Components

The following are Cisco TSP components:

- CiscoTSP dll– TAPI service implementation provided by Cisco TSP
- CTIQBE over TCP/IP – Cisco protocol used to monitor and control devices and lines
- CTI Manager Service – Manages CTI resources and connections to devices. Exposed to 3rd-party applications via Cisco TSP and/or JTAPI API

## Cisco Media Drivers

Cisco Media Driver can be used to play announcements or record the call media. For information about the installation of the Media Drivers, see [Cisco Media Driver Selection, page 4-5](#).

## TAPI Debugging

The TAPI browser is a TAPI debugging application. It can be downloaded from the Microsoft MSDN Web site at <ftp://ftp.microsoft.com/developr/TAPI/tb20.zip>. The TAPI browser can be used to initialize TAPI, for use by TAPI developers to test a TAPI implementation and to verify that the TSP is operational.

## CTI Manager (Cluster Support)

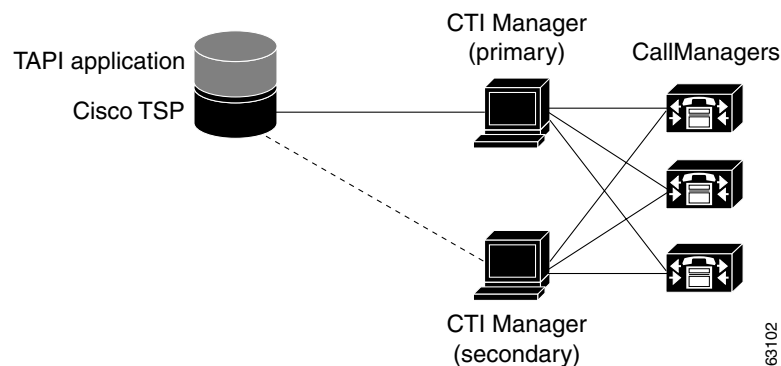
The CTI Manager, along with the Cisco Unified TSP, provide an abstraction of the Cisco Unified Communications Manager cluster that allows TAPI applications to access Cisco Unified Communications Manager resources and functionality without being aware of any specific Cisco Unified Communications Manager. The Cisco Unified Communications Manager cluster abstraction also enhances the failover capability of CTI Manager resources. A failover condition occurs when a node fails, a CTI Manager fails, or a TAPI application fails, as illustrated in [Figure 1-2](#).



### Note

Cisco does not support CTI device monitoring or call control with 3rd-party devices.

**Figure 1-2 Cluster Support Architecture**



## Cisco Unified Communications Manager Failure

When a Cisco Unified Communications Manager node in a cluster fails, the CTI Manager recovers the affected CTI ports and route points by reopening these devices on another Cisco Unified Communications Manager node. When the failure is first detected, Cisco Unified TSP sends a PHONE\_STATE (PHONESTATE\_SUSPEND) message to the TAPI application.

When the CTI port/route point is successfully reopened on another Cisco Unified Communications Manager, Cisco Unified TSP sends a phone PHONE\_STATE (PHONESTATE\_RESUME) message to the TAPI application. If no Cisco Unified Communications Manager is available, the CTI Manager waits until an appropriate Cisco Unified Communications Manager comes back in service and tries to open the device again. The lines on the affected device also go out of service and in service with the corresponding

LINE\_LINEDEVSTATE (LINEDEVSTATE\_OUTOFSERVICE) and LINE\_LINEDEVSTATE (LINEDEVSTATE\_INSERVICE) events Cisco Unified TSP sends to the TAPI application. If for some reason the device or lines cannot be opened, even when all Cisco Unified Communications Managers come back in service, the system closes the devices or lines, and Cisco Unified TSP will send PHONE\_CLOSE or LINE\_CLOSE messages to the TAPI application.

When a failed Cisco Unified Communications Manager node comes back in service, CTI Manager “re-homes” the affected CTI ports or route points to their original Cisco Unified Communications Manager. The graceful re-homing process ensures that the re-homing only starts when calls are no longer being processed or are active on the affected device. For this reason, the re-homing process may not finish for a long time, especially for route points, which can handle many simultaneous calls.

When a Cisco Unified Communications Manager node fails, phones currently re-home to another node in the same cluster. If a TAPI application has a phone device opened and the phone goes through the re-homing process, CTI Manager automatically recovers that device, and Cisco Unified TSP sends a PHONE\_STATE (PHONESTATE\_SUSPEND) message to the TAPI application. When the phone successfully re-homes to another Cisco Unified Communications Manager node, Cisco Unified TSP sends a PHONE\_STATE (PHONESTATE\_RESUME) message to the TAPI application.

The lines on the affected device also go out of service and in service, and Cisco Unified TSP sends LINE\_LINEDEVSTATE (LINEDEVSTATE\_OUTOFSERVICE) and LINE\_LINEDEVSTATE (LINEDEVSTATE\_INSERVICE) messages to the TAPI application.

## Call Survivability

When a device or Cisco Unified Communications Manager failure occurs, no call survivability exists; however, media streams that are already connected between devices will survive. Calls in the process of being set up or modified (transfer, conference, redirect) simply get dropped.

## CTI Manager Failure

When a primary CTI Manager fails, Cisco Unified TSP sends a PHONE\_STATE (PHONESTATE\_SUSPEND) message and a LINE\_LINEDEVSTATE (LINEDEVSTATE\_OUTOFSERVICE) message for every phone and line device that the application opened. Cisco Unified TSP then connects to a backup CTI Manager. When a connection to a backup CTI Manager is established and the device or line successfully reopens, the Cisco Unified TSP sends a PHONE\_STATE (PHONESTATE\_RESUME) or LINE\_LINEDEVSTATE (LINEDEVSTATE\_INSERVICE) message to the TAPI application. If the Cisco Unified TSP is unsuccessful in opening the device or line for a CTI port or route point, the Cisco Unified TSP closes the device or line by sending the appropriate PHONE\_CLOSE or LINE\_CLOSE message to the TAPI application.

After Cisco Unified TSP is connected to the backup CTI Manager, Cisco Unified TSP will not reconnect to the primary CTI Manager until the connection is lost between Cisco Unified TSP and the backup CTI Manager.

If devices are added to or removed from the user while the CTI Manager is down, Cisco Unified TSP generates PHONE\_CREATE/LINE\_CREATE or PHONE\_REMOVE/LINE\_REMOVE events, respectively, when connection to a backup CTI Manager is established.



## Cisco Unified TAPI Application Failure

When a Cisco TAPI application fails (the CTI Manager closes the provider), calls at CTI ports and route points that have not yet been terminated get redirected to the Call Forward On Failure (CFF) number that has been configured for them. The system routes new calls into CTI Ports and Route Points that are not opened by an application to their CFNA number.

## LINE\_CALLDEVSPECIFIC Event Support for RTP Events

RTP events are generated as LINE\_CALLDEVSPECIFIC events that contain Call Handle details of the call. However, to activate the feature, the application must negotiate the extension version greater than or equal to 0x00040001 when opening the line.

Due to dependency on the extension version of the line, the Media Events, RTP\_START / STOP, are reported differently to the application:

- If extension version is less than EXTVERSION\_FOUR\_DOT\_ZERO - 0x00040000 — TSP reports LINE\_DEVSPECIFIC event to application on the line irrespective whether call object is present. In this case, even if a call is DeAllocated after IDLE state, RTP\_STOP events are delivered to the application.
- If extension version is greater than or same as EXTVERSION\_FOUR\_DOT\_ZERO - 0x00040000—TSP does report the Media Events if the Call Object is DeAllocated from Application.

So a check must be added for the Extension Version to maintain backward compatibility. So it must not be assumed that RTP events will always come before IDLE event.

## QoS

Cisco Unified TSP supports the Cisco baseline for baselining of Quality of Service (QoS). Cisco Unified TSP marks the IP Differentiated Services Code Point (DSCP) for QBE control signals that flow from TSP to CTI with the value of the Service parameter “DSCP IP for CTI Applications” that CTI sends in the ProviderOpenCompletedEvent. The Cisco TAPI Wave driver marks the RTP packets with the value that CTI sends in the StartTransmissionEvent. The system stores the DSCP value received in the StartTransmissionEvent in the DevSpecific portion of the LINECALLINFO structure, and fires the LINECALLINFOSTATE\_DEVSPECIFIC event with the QoS indicator.

**Note**

QoS information is not available if you begin monitoring in the middle of a call because existing calls do not have an RTP event.

## Presentation Indication (PI)

There is a need to separate the presentability aspects of a number (calling, called, and so on) from the actual number itself. For example, when the number is not to be displayed on the IP phone, the information might still be needed by another system, such as Unity VM. Hence, each number/name of the display name needs to be associated with a Presentation Indication (PI) flag, which will indicate whether the information should be displayed to the user or not.

You can set up this feature as follows:

#### On a Per-Call Basis

You can use Route Patterns and Translation Patterns to set or reset PI flags for various partyDNs/Names on a per-call basis. If the pattern matches the digits, the PI settings that are associated with the pattern will be applied to the call information.

#### On a Permanent Basis

You can configure a trunk device with “Allow” or “Restrict” options for parties. This will set the PI flags for the corresponding party information for all calls from this trunk.

Cisco Unified TSP supports this feature. If calls are made via Translation patterns with all of the flags set to Restricted, the system sends the CallerID/Name, ConnectedID/Name, and RedirectionID/Name to applications as Blank. The system also sets the LINECALLPARTYID flags to Blocked if both the Name and Party number are set to Restricted.

When developing an application, be sure only to use functions that the Cisco TAPI Service Provider supports. For example, the Cisco TAPI Service Provider supports transfer, but not fax detection. If an application requires an unsupported media or bearer mode, the application will not work as expected.

Cisco Unified TSP does not support TAPI 3.0 applications.

## Call Control

You can configure Cisco Unified TSP to provide first- or third-party call control.

### First-Party Call Control

In first-party call control, the application terminates the audio stream. Ordinarily, this occurs by using the Cisco wave driver. However, if you want the application to control the audio stream instead of the wave driver, use the Cisco device-specific extensions.

### Third-Party Call Control

In third-party call control, the control of an audio stream terminating device is not “local” to the Cisco Unified Communications Manager. In such cases, the controller might be the physical IP phone on your desk or a group of IP phones for which your application is responsible.

**Note**

---

Cisco does not support CTI device monitoring or call control with 3rd-party devices.

---

## CTI Port

For first-party call control, a CTI port device must exist in the Cisco Unified Communications Manager. Because each port can only have one active audio stream at a time, most configurations only need one line per port.

A CTI port device does not actually exist in the system until you run a TAPI application and a line on the port device is opened requesting LINEMEDIAMODE\_AUTOMATEDVOICE and LINEMEDIAMODE\_INTERACTIVEVOICE. Until the port is opened, anyone who calls the directory number that is associated with that CTI port device receives a busy or reorder tone.

The IP address and UDP port number is either specified statically (the same IP address and UDP port number is used for every call) or dynamically. By default, CTI ports use static registration.

## Dynamic Port Registration

Dynamic Port Registration enables applications to specify the IP address and UDP port number on a call-by-call basis. Currently, the IP address and UDP port number are specified when a CTI port registers and is static through the life of the registration of the CTI port. When media is requested to be established to the CTI port, the system uses the same static IP address and UDP port number for every call.

An application that wants to use Dynamic Port Registration must specify the IP address and UDP port number on a call before invoking any features on the call. If the feature is invoked before the IP address and UDP port number are set, the feature will fail, and the call state will be set depending on when the media time-out occurs.

## CTI Route Point

You can use Cisco Unified TAPI to control CTI route points. CTI route points allow Cisco Unified TAPI applications to redirect incoming calls with an infinite queue depth. This allows incoming calls to avoid busy signals.

CTI route point devices have an address capability flag of `LINEADDRCAPFLAGS_ROUTEPOINT`. When your application opens a line of this type, it can handle any incoming call by disconnecting, accepting, or redirecting the call to some other directory number. The basis for redirection decisions can be caller ID information, time of day, or other information that is available to the program.

## Media Termination at Route Point

The Media Termination at Route Point feature lets applications terminate media at route points. This feature enables applications to pass the IP address and port number where they want the call at the route point to have media established.

The system supports the following features at route points:

- Answer
- Multiple Active Calls
- Redirect
- Hold
- UnHold
- Blind Transfer
- DTMF Digits
- Tones

## Monitoring Call Park Directory Numbers

The Cisco Unified TSP supports monitoring calls on lines that represent Call Park Directory Numbers (Call Park DN). The Cisco Unified TSP uses a device-specific extension in the LINEDEVCAPS structure that allows TAPI applications to differentiate Call Park DN lines from other lines. If an application opens a Call Park DN line, all calls that are parked to the Call Park DN get reported to the application. The application cannot perform any call control functions on any calls at a Call Park DN.

To open Call Park DN lines, you must check the **Monitor Call Park DN**s check box in Cisco Unified Communications Manager User Administration for the Cisco Unified TSP user. Otherwise, the application will not perceive any of the Call Park DN lines upon initialization.

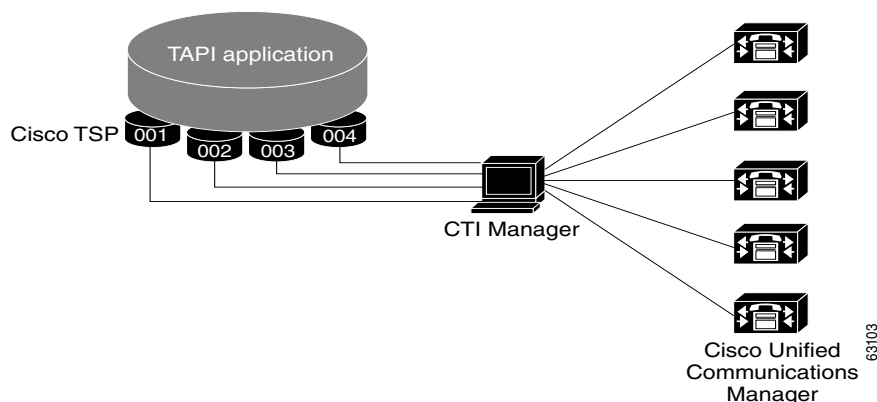
## Multiple Cisco Unified TSPs

In the Cisco Unified TAPI solution, the TAPI application and Cisco Unified TSP get installed on the same machine. The Cisco Unified TAPI application and Cisco Unified TSP do not directly interface with each other. A layer written by Microsoft sits between the TAPI application and Cisco Unified TSP. This layer, known as TAPISRV, allows the installation of multiple TSPs on the same machine, and it hides that fact from the Cisco Unified TAPI application. The only difference to the TAPI application is that it is now informed that there are more lines that it can control.

Consider an example—assume that Cisco Unified TSP1 exposes 100 lines, and Cisco Unified TSP2 exposes 100 lines. In the single Cisco Unified TSP architecture where Cisco Unified TSP1 is the only Cisco Unified TSP that is installed, Cisco Unified TSP1 would tell TAPISRV that it supports 100 lines, and TAPISRV would tell the application that it can control 100 lines. In the multiple Cisco Unified TSP architecture, where both Cisco Unified TSPs are installed, this means that Cisco Unified TSP1 would tell TAPISRV that it supports 100 lines, and Cisco Unified TSP2 would tell TAPISRV that it supports 100 lines. TAPISRV would add the lines and inform the application that it now supports 200 lines. The application communicates with TAPISRV, and TAPISRV takes care of communicating with the correct Cisco Unified TSP.

Ensure that each Cisco Unified TSP is configured with a different username and password that you administer in the Cisco Unified Communications Manager Directory. Configure each user in the Directory, so devices that are associated with each user do not overlap. Each Cisco Unified TSP in the multiple Cisco Unified TSP system does not communicate with the others. Each Cisco Unified TSP in the multiple Cisco Unified TSP system creates a separate CTI connection to the CTI Manager as shown in [Figure 1-3](#). Multiple Cisco Unified TSPs help in scalability and higher performance.

**Figure 1-3 Multiple Cisco Unified TSPs Connect to CTI Manager**



## CTI Device/Line Restriction

With CTI Device/Line restriction implementation, a CTIRestricted flag is placed on device or line basis. When a device is restricted, it assumes that all its configured lines are restricted.

Cisco Unified TSP does not report any restricted devices and lines back to application. When a CTIRestricted flag is changed from Cisco Unified Communications Manager Administration, Cisco Unified TSP treats it as normal device/line add or removal.

## Development Guidelines

Cisco maintains a policy of interface backward compatibility for at least one previous major release of Cisco Unified Communications Manager. Cisco still requires Cisco Technology Developer Program member applications to be retested and updated as necessary to maintain compatibility with each new major release of Cisco Unified Communications Manager.

The following practices are recommended to all developers, including those in the Cisco Technology Developer Program, to reduce the number and extent of any updates that may be necessary:

- The order of events and/or messages may change. Developers should not depend on the order of events or messages. For example, where a feature invocation involves two or more independent transactions, the events or messages may be interleaved. Events related to the second transaction may precede messages related to the first. Additionally, events or messages can be delayed due to situations beyond control of the interface (for example, network or transport failures). Applications should be able to recover from out of order events or messages, even when the order is required for protocol operation.
- The order of elements within the interface event or message may change, within the constraints of the protocol specification. Developers must avoid unnecessary dependence on the order of elements to interpret information.
- New interface events, methods, responses, headers, parameters, attributes, other elements, or new values of existing elements, may be introduced. Developers must disregard or provide generic treatments where necessary for any unknown elements or unknown values of known elements encountered.
- Previous interface events, methods, responses, headers, parameters, attributes, and other elements, will remain, and will maintain their previous meaning and behavior to the extent possible and consistent with the need to correct defects.
- Applications must not be dependent on interface behavior resulting from defects (behavior not consistent with published interface specifications) since the behavior can change when defect is fixed.
- Use of deprecated methods, handlers, events, responses, headers, parameters, attributes, or other elements must be removed from applications as soon as possible to avoid issues when those deprecated items are removed from Cisco Unified Communications Manager.
- Application Developers must be aware that not all new features and new supported devices (for example, phones) will be forward compatible. New features and devices may require application modifications to be compatible and/or to make use of the new features/devices.

