

# Comprender la arquitectura JTAPI y el flujo de llamadas en UCCX

## Contenido

---

### [Introducción](#)

### [Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

### [Antecedentes](#)

[Descripción general de JTAPI](#)

[JTAPI y UCCX](#)

[Arquitectura JTAPI](#)

[Modelo JTAPI Observer](#)

[Modelo de proveedor JTAPI](#)

[Usuario JTAPI](#)

[Identificadores P1 y P2](#)

[Flujo de llamada](#)

### [Troubleshoot](#)

[Habilitar y recopilar registros JTAPI](#)

[Habilitar depuraciones JTAPI](#)

[Recopilar depuraciones de JTAPI](#)

[Desde RTMT](#)

[Desde CLI](#)

[ubicación del registro JTAPI](#)

### [Enlaces de referencia](#)

---

## Introducción

Este documento describe la funcionalidad básica de JTAPI, su arquitectura y el flujo de llamadas con respecto a UCCX.

## Prerequisites

Requirements

Cisco recomienda conocer estas herramientas y funciones:

- JTAPI: API de telefonía Java
- API: interfaz de programación de aplicaciones

- UCCX: Unified Contact Center Express
- CUCM: Cisco Unified Communications Manager
- CTI - Integración de telefonía y ordenador

Cisco recomienda conocer estos temas:

- Configuración de Cisco Unified Communications Manager (CUCM)
- Configuración de Unified Contact Center Express (UCCX)

#### Componentes Utilizados

La información que contiene este documento se basa en estas versiones de software:

- UCCX versión 12.5.1.11002-481
- CUCM, versión 12.5.1.11900-146

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

#### Antecedentes

Este documento describe la arquitectura JTAPI, el flujo de llamadas, la habilitación de las depuraciones y la recopilación de los registros.

#### Descripción general de JTAPI

Cisco Unified JTAPI actúa como un estándar de interfaz de programación desarrollado por Sun Microsystems para su uso con aplicaciones de telefonía y ordenador basadas en Java. Cisco JTAPI implementa la especificación Sun JTAPI 1.2 con extensiones adicionales de Cisco. Debe utilizar Cisco JTAPI para desarrollar aplicaciones que:

- 

Controle y observe los teléfonos de Cisco Unified Communications Manager.

- 

Enrutar llamadas mediante puertos de integración de telefonía e informática (CTI) y puntos de ruta (dispositivos virtuales).

## JTAPI y UCCX

Cisco Unified JTAPI se utiliza en un centro de contacto para supervisar el estado del dispositivo y para emitir instrucciones de enrutamiento para enviar llamadas al lugar adecuado en el momento adecuado. Además, JTAPI se utiliza para iniciar y detener las instrucciones de grabación mientras se recuperan las estadísticas de llamadas para su análisis y para filtrar las llamadas emergentes en aplicaciones CRM, secuencias de comandos automatizadas y control remoto de llamadas.

## Arquitectura JTAPI

Cisco Unified JTAPI, que se utiliza en un entorno empresarial, combina la disponibilidad de los usuarios, la ubicación y las preferencias para crear un entorno personalizado exclusivo para el routing basado en presencia.

Estas son las aplicaciones de JTAPI:

- JTAPI puede supervisar o recibir notificaciones sobre dos o más combinaciones de teléfono y línea.
- Las aplicaciones del centro de contacto utilizan el modelo de llamada completa para JTAPI.
- JTAPI proporciona esta funcionalidad:
  - Control de llamadas
  - Control de medios
  - Negociación de medios

## Modelo JTAPI Observer

En el siguiente diagrama se explica el modelo Provider-Observer en el que funciona JTAPI.

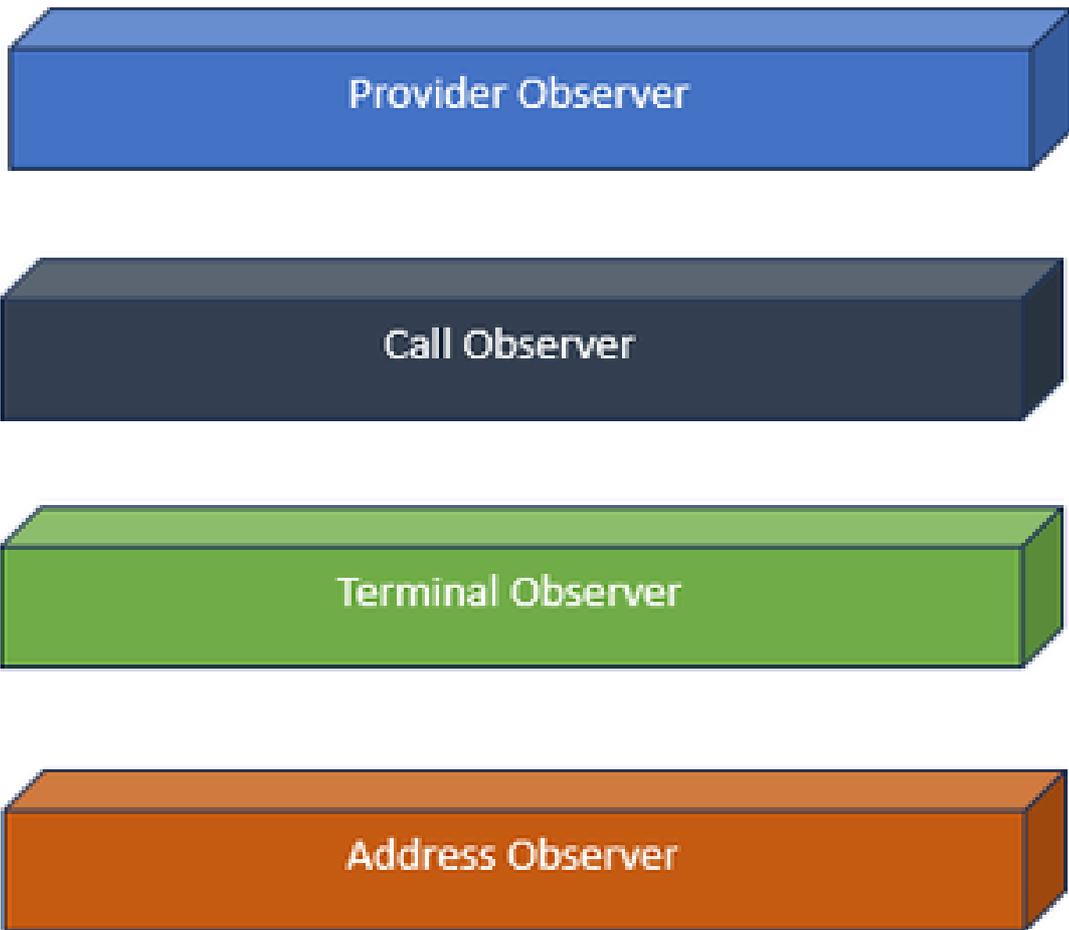
Provider

Call

Terminal

Address

*Observadores*



### *Interfaces de Observer*

JTAPI se basa en la idea de observador. Por observador, se refiere a la idea del que observa los acontecimientos. Puede colocar varios observadores en diferentes elementos del sistema. JTAPI utiliza observadores para conocer los cambios de estado de los objetos.

Por ejemplo, puede hacer que se coloquen observadores en las líneas, teléfonos, terminales, direcciones, etc., y obtener información sobre sus cambios de estado.

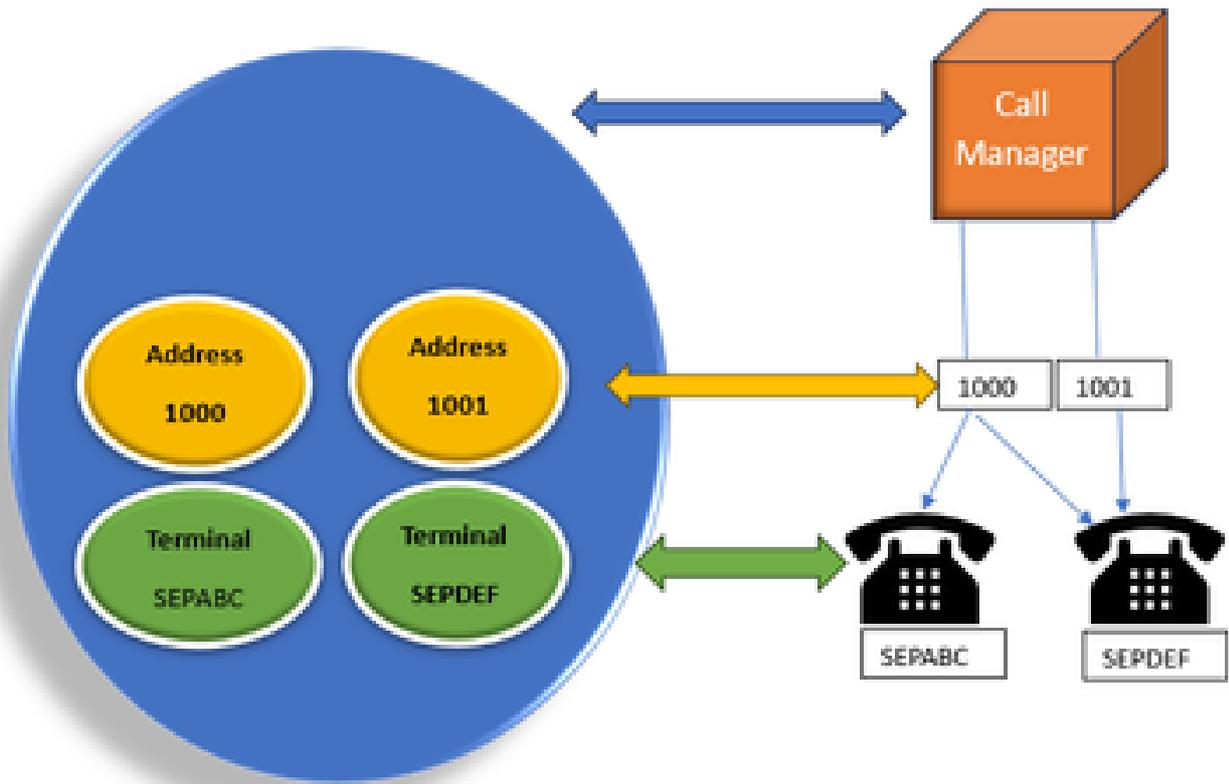


**Nota:** Si no tiene observadores colocados en un objeto, no podrá recibir notificaciones sobre las acciones que se llevan a cabo con ellos.

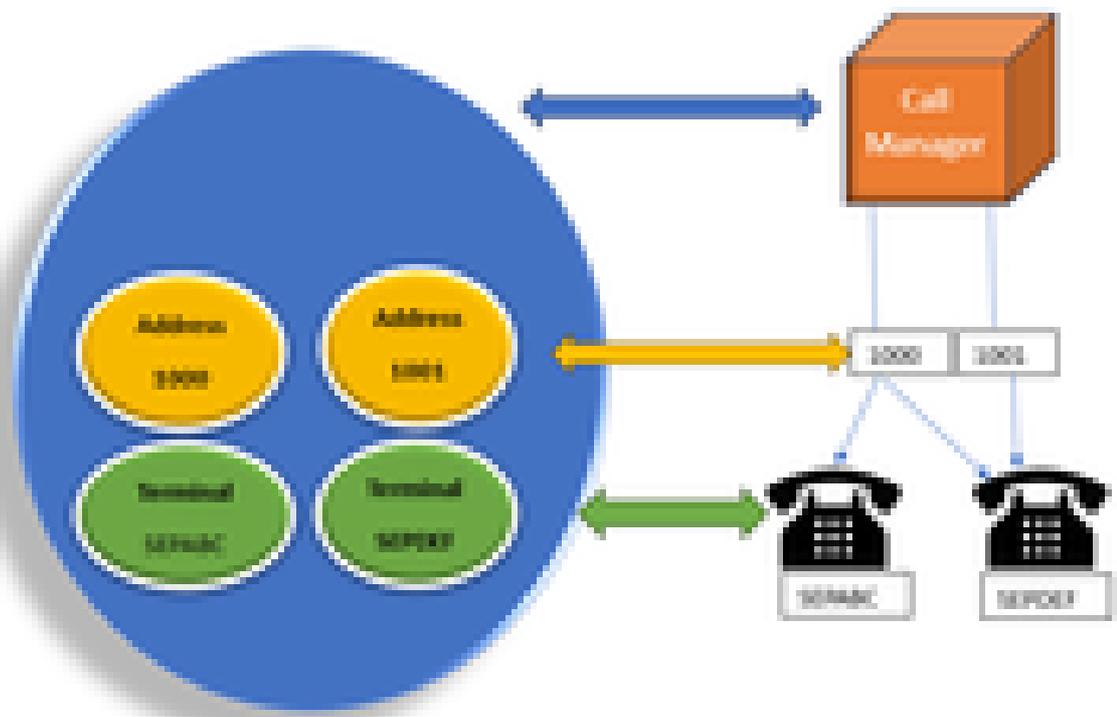
---

## Modelo de proveedor JTAPI

En el siguiente diagrama se explica el modelo de proveedor en el que funciona JTAPI:



Modelo



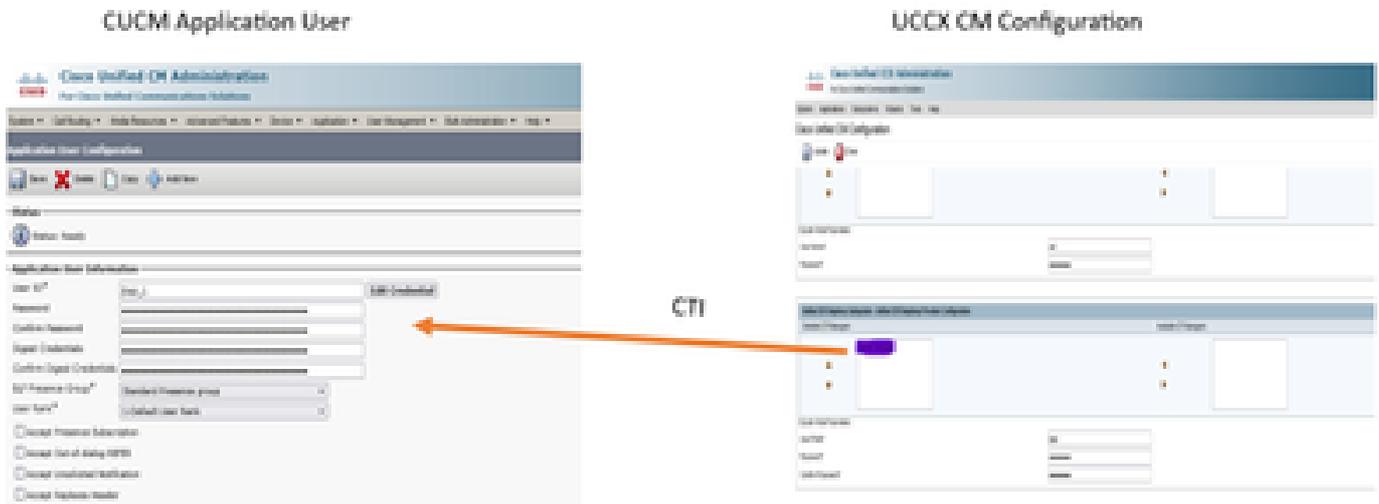
de proveedor  
de JTAPI modelo de proveedor

El proveedor JTAPI es una conexión de la aplicación en Call Manager o CTI Manager. Los proveedores se utilizan para adjuntar observadores a los objetos. También puede asociar un observador a un proveedor. Los proveedores se utilizan para recibir notificaciones sobre las acciones

realizadas en los objetos. (También puede tomar el control de los dispositivos en los que está conectado el observador (desde el proveedor que conectó ese observador).

## Usuario JTAPI

Las siguientes capturas de pantalla se toman de CUCM (izquierda) y UCCX (derecha) que explican el usuario de la aplicación JTAPI.



- Cuando inicia una aplicación y desea establecer una conexión con el administrador de CTI, abre un proveedor.
- El motivo para abrir un proveedor es que puede supervisar las acciones realizadas en los dispositivos, terminales, etc.
- La forma en que se implementa en CUCM es a través de un usuario de la aplicación.
- Cree este usuario y proporcione credenciales para que pueda autenticarse primero en CTI en CUCM.
- Por lo tanto, el usuario de la aplicación JTAPI que se crea en CUCM es el proveedor.
- Aparte de solo monitorear, debe asegurarse de que esos dispositivos estén bajo **Dispositivos asociados** para asegurarse de que puede controlar los dispositivos.



**Nota:** Si no crea el usuario JTAPI en cucm, lo crea la aplicación (UCCX) mediante AXL en CUCM.

---

#### Identificadores P1 y P2

P1 y P2 son los identificadores de proveedor. Esto es importante cuando va a abrir varios proveedores desde la misma aplicación. Desde UCCX, se crean 2 proveedores, uno de los cuales controla los puertos CTI y los puntos de ruta, y el otro controla los teléfonos del agente llamados proveedor RM-JTAPI. Usted, como UCCX, crea el proveedor que controla los puertos CTI y los puntos de ruta en primer lugar, que es el proveedor JTAPI P1. El otro proveedor que crea después de P1 es el proveedor P2 o el proveedor RmCm que gestiona los dispositivos de agente.

Si ve un evento de llamada nueva P1, sabrá que es un evento de llamada desde un puerto CTI o un punto de ruta CTI. Si ve el evento de nueva llamada P2, sabrá que se trata de un evento de llamada del teléfono real. Se crea un usuario RM-JTAPI y un usuario JTAPI en el lado CCX, pero

en el lado CUCM, se ven 2 usuarios JTAPI creados. Esto se debe a que cada nodo CCX obtiene su propio usuario JTAPI, pero comparten el usuario RM-JTAPI. Al crear un desencadenador en UCCX, se agrega a ambos usuarios de JTAPI. Ambos nodos abren un proveedor por separado. Por lo tanto, cualquier acción realizada en el punto de ruta se notifica en ambos nodos CCX.

Aparte de eso, el nodo secundario simplemente se sienta allí y sigue informando que sigue siendo el nodo secundario. Cada nodo tiene un grupo independiente de puertos CTI. Los puertos CTI del nodo 1 son controlados por jtapi\_1. Los puertos CTI del nodo 2 son controlados por jtapi\_2.

Cuando la llamada llega al punto de ruta CTI, se notifica a ambos nodos CCX sobre el nuevo evento de llamada, pero el nodo CCX activo responde al Call Manager con una solicitud de redirección para uno de sus puertos CTI que controla.

Si ve una IP contra un punto de ruta CTI en CUCM, es una de las IP de CCX, pero eso no significa que un nodo específico esté enrutando la llamada.

Una de las acciones más comunes es desasociar el dispositivo del usuario JTAPI y volver a agregarlo. La razón detrás de esto es que cuando se anula la asociación, el proveedor recibe una notificación al respecto y elimina todas las conexiones a él y luego, cuando se vuelve a agregar, el observador se agrega de nuevo y el proveedor comienza a monitorearlo de nuevo mediante una solicitud de proveedor abierto. Puede ver que aparte del dispositivo que se agrega en la lista de dispositivos controlados, hay otra cosa llamada casilla de verificación **Permitir el control del dispositivo desde CTI** en el dispositivo individual también. Esto es para aportar un nivel adicional de granularidad. Por lo tanto, si tiene el punto de ruta agregado en la lista controlada pero la casilla de verificación CTI no está marcada, puede recibir notificaciones sobre los eventos en ese punto de ruta pero no es posible realizar ninguna acción en el control de llamadas.

#### Flujo de llamada

A continuación se indican los sucesivos eventos relacionados con el flujo de llamadas de UCCX:

- Cuando la llamada llega al punto de ruta CTI, se dirige a un puerto CTI.
- El puerto CTI abre el canal de medios con el controlador ipvms en la caja uccx.
- Una vez que decide que el agente debe atender la llamada, se realiza una transferencia con consulta desde el puerto al agente.
- El nuevo evento de llamada se envía al punto de ruta CTI.
- La solicitud de redirección va al puerto CTI.
- El estado del identificador de llamada pasa a ser inactivo.
- A continuación, se produce otro nuevo evento de llamada para la llamada al puerto CTI.
- Si la respuesta de redirección es 0, es buena, si no es cero, significa que falló.
- A continuación, envía una solicitud de aceptación de llamada (esta solicitud no se contesta, solo se acepta en el puerto).
- A continuación, acepte resultados de paso en el script, esto es cuando la solicitud de respuesta de llamada entra en Jtapi.



**Nota:** Esto sucede muy rápido y a veces ocurren varios eventos al mismo tiempo, como llamadas provenientes de Cisco Unity Connection o transferencias que toman tiempo de CUCM, esto puede causar una condición RACE en el paso de aceptación, que es la razón por la que desea ralentizar esto. Para ello, agregue un paso de retraso antes de aceptar el paso.

---

11. Luego obtienes una respuesta del puerto.

12. El estado de la llamada cambia a conectado.

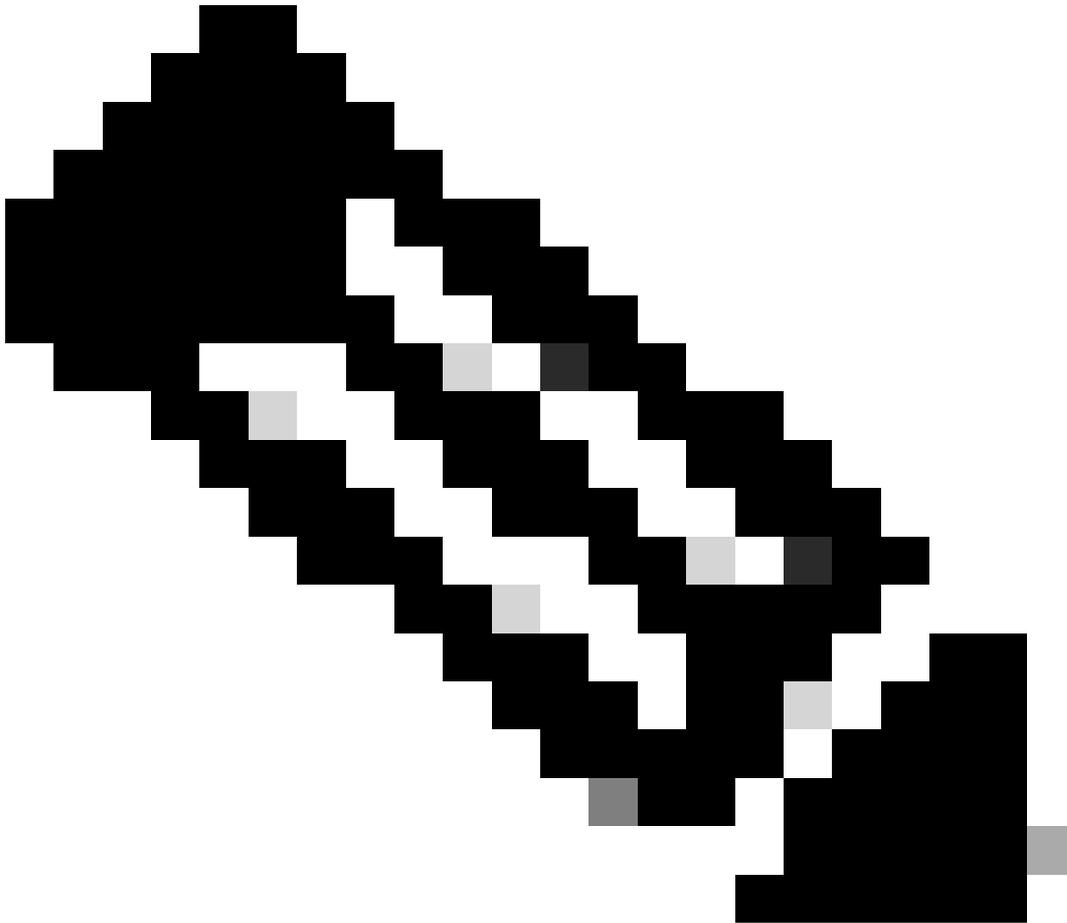


**Nota:** CTI es asíncrono a diferencia de los teléfonos sip o skinny que esperan la respuesta antes de enviar una nueva solicitud, por lo que el orden de los mensajes en los mensajes JTAPI CTI se puede confundir.

- 
13. Después de obtener la respuesta de respuesta del puerto, se produce la negociación de medios.
  14. El puerto envía una solicitud de canal\_lógico\_abierto en la que comparte su puerto e ip a los que desea que la parte remota envíe el RTP.
  15. Antes de abrir el canal lógico, crea una conexión con el controlador IPVMS en el cuadro UCCX y abre una secuencia RTP.
  16. El siguiente evento es start\_receive, que nos indica la información del otro extremo (es decir, la dirección IP y el puerto del dispositivo de llamada).
  17. Entonces usted consigue start\_transmission evento como cualquier otra señal de medios.

18. Ahora está escuchando la IVR y las indicaciones.

---



**Nota:** Aquí es donde finaliza la configuración de la llamada.

---

19. Ahora, cuando la llamada llega a un paso en el script que permite que la llamada se transfiera al agente, se ve un CallSetupTransferRequest.

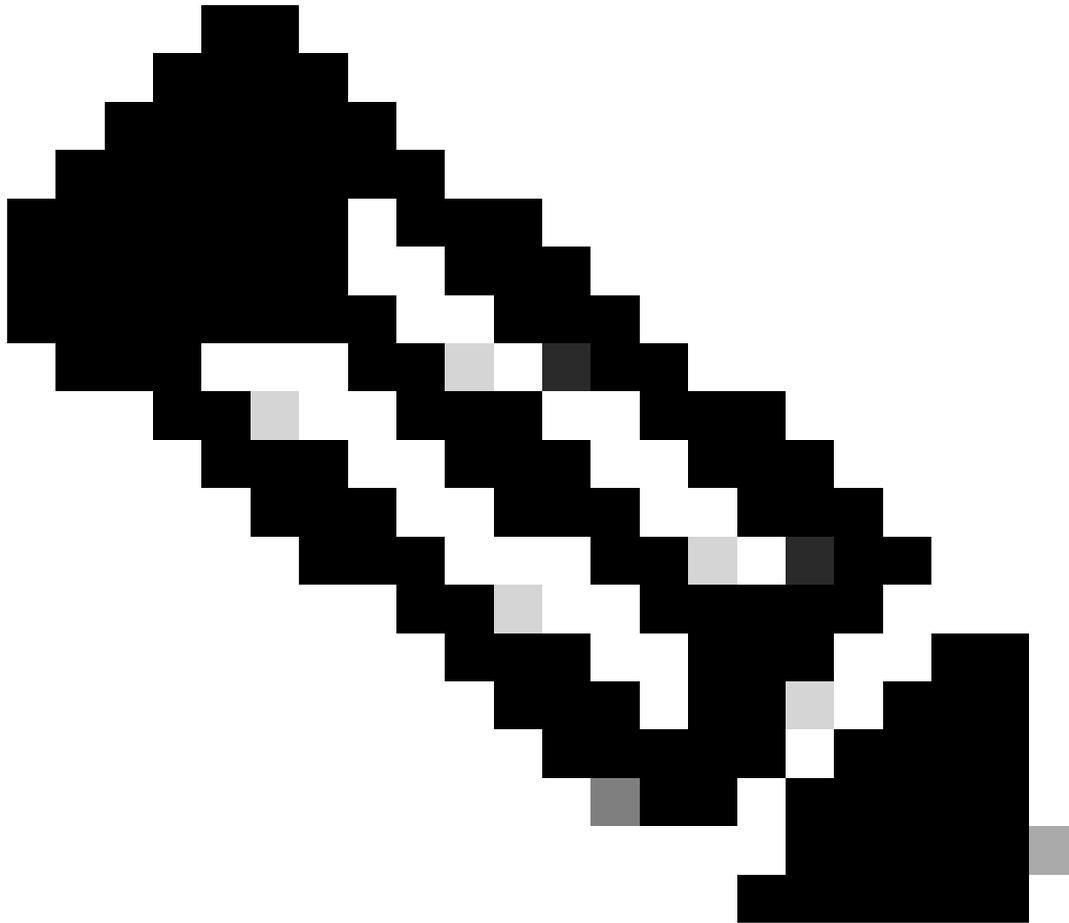
20. A diferencia del primer mensaje, se trata de una transferencia con consulta y no de una redirección.

21. La razón de que esto sea una transferencia con consulta es porque si un agente está LISTO pero no en su asiento, y redirigimos la llamada, simplemente permanece sin responder, pero si hacemos una transferencia con consulta, entonces si el agente no está allí, entonces la llamada nuevamente se coloca en la cola.

22. Al igual que cualquier otra transferencia con consulta, este es el puerto CTI que golpea el botón de transferencia por primera vez en el

teléfono.

---



**Nota: ConsultWithoutMedia** es la API para la transferencia con consulta.

---

23. En una transferencia de consulta regular, existe una ruta de medios entre A y C, pero en este caso se indica a CUCM que no lo haga, ya que no tiene sentido que establezca medios entre UCCX y el agente.
24. Así que va a hacer una transferencia de consulta sin hacer un corte de medios entre el agente y el puerto CTI.
25. En este punto, el puerto CTI pone a la persona que llama en espera y vemos un evento de cambio de estado de llamada=EN ESPERA.
26. Ahora verá un evento newcall desde el puerto CTI al dispositivo del agente.(Usando el id de llamada original, pero un subconjunto de él y un

evento P2.)

27. Si busca el ID de llamada de evento P2, verá el siguiente mensaje como solicitud de respuesta de llamada, lo que significa que el agente atendió la llamada.

28. Una vez que sepa que el agente ha atendido la llamada (mediante P2) y que la llamada también está en estado conectado en el lado del puerto CTI (mediante P1), verá un punto de ruta CallDirectTransferRequest, lo que significa que el botón de transferencia se ha pulsado por segunda vez.

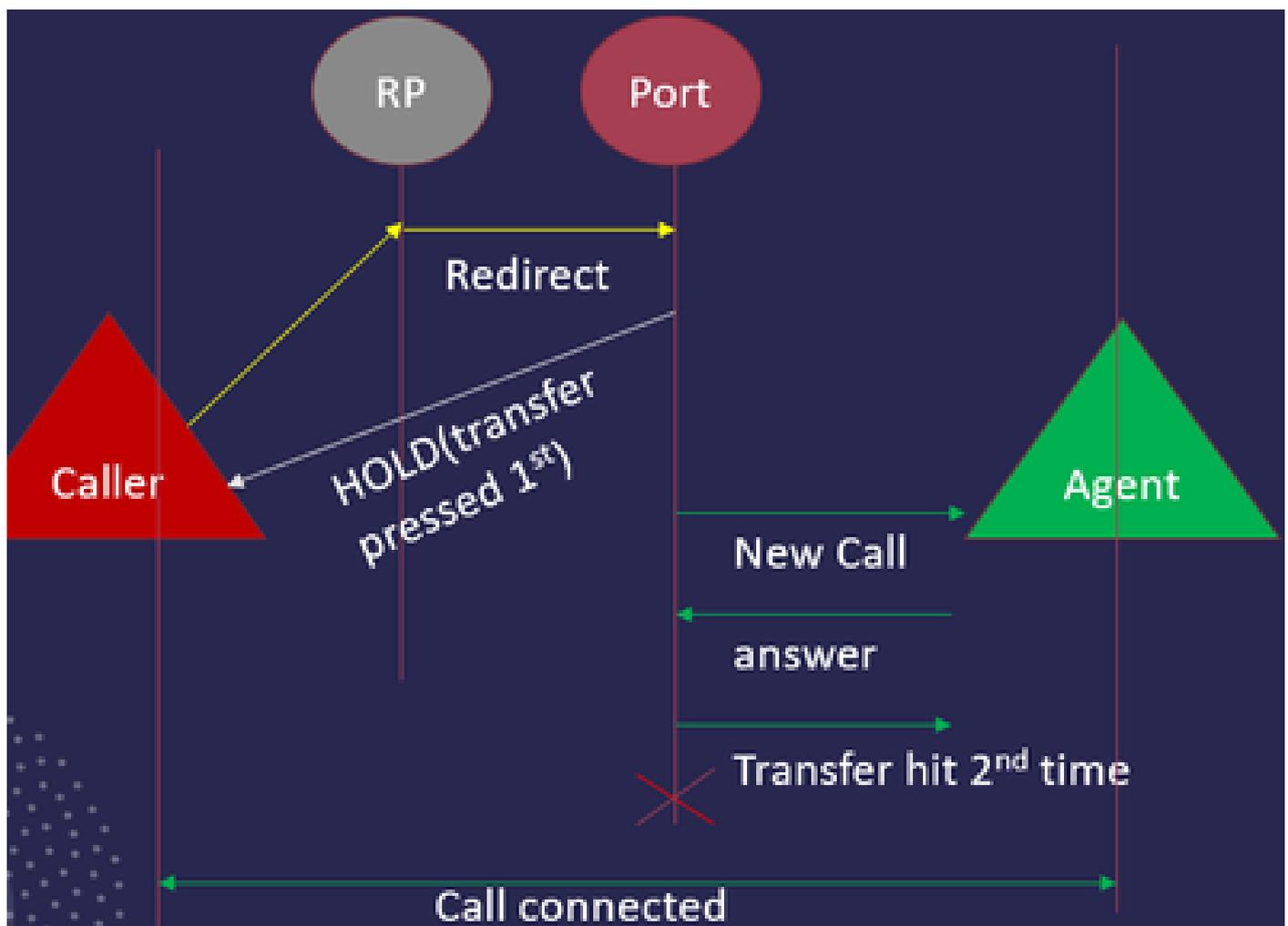
29. Ahora que el puerto CTI sabe que el agente ha atendido la llamada, no tiene sentido esperar, envía inmediatamente un mensaje CallDirectTransferRequest que es el puerto CTI que presiona el botón de transferencia por segunda vez, como se explicó anteriormente.

30. Ahora, el tramo de llamada original (el que estaba en espera) es el que sobrevive.

31. El otro tramo de llamada se destruye (entre puerto y agente).

32. En este momento, CUCM y UCCX salen de la imagen y RTP se establece entre la persona que llama y el agente a través de la puerta de enlace.

El siguiente diagrama explica el flujo de llamadas mencionado anteriormente de forma resumida.



Resumen de flujo de llamadas JTAPI

Habilitar y recopilar registros JTAPI

Habilitar depuraciones JTAPI

Verifique estos pasos para habilitar los niveles de depuración JTAPI.

- Inicie sesión en UCCX.
- Vaya a **Mantenimiento de CCX**.
- Vaya a **Seguimiento**.
- Vaya a **Configuración**.
- En la lista desplegable Seleccionar servicio, seleccione **Cliente de telefonía Cisco Unified CM**.
- Seleccione la casilla de verificación **Depuración**.
- Seleccione todas las casillas de verificación excepto **MISC\_DEBUGGING**.

Recopilar depuraciones de JTAPI

Verifique estos pasos para habilitar los niveles de depuración JTAPI desde RTMT o CLI.

Desde RTMT

- Inicie sesión en la herramienta de supervisión en tiempo real de CCX.
- Haga clic en **Trace & Log Central**.
- Haga clic en **Collect Files**.
- Seleccione **Cliente JTAPI** para todos los servidores.
- Haga clic en Next (Siguiente).
- Seleccione las marcas de tiempo y la ubicación en la que desea guardar los archivos de registro.

Desde CLI

ubicación del registro JTAPI

activelog /uccx/log/JTAPI

Comando para recopilar los registros de JTAPI:

```
file get activelog /uccx/log/JTAPI/* recurs compress
```

**Sintaxis:**

```
file get {activelog|inactivelog|install} file-spec [options]
```

file-spec archivo obligatorio para transferir

options optional reltime meses|semanas|días|horas|minutos timevalue

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

match regex

recurs

comprimir

5 maneras de descargar los registros basados en la marca de tiempo

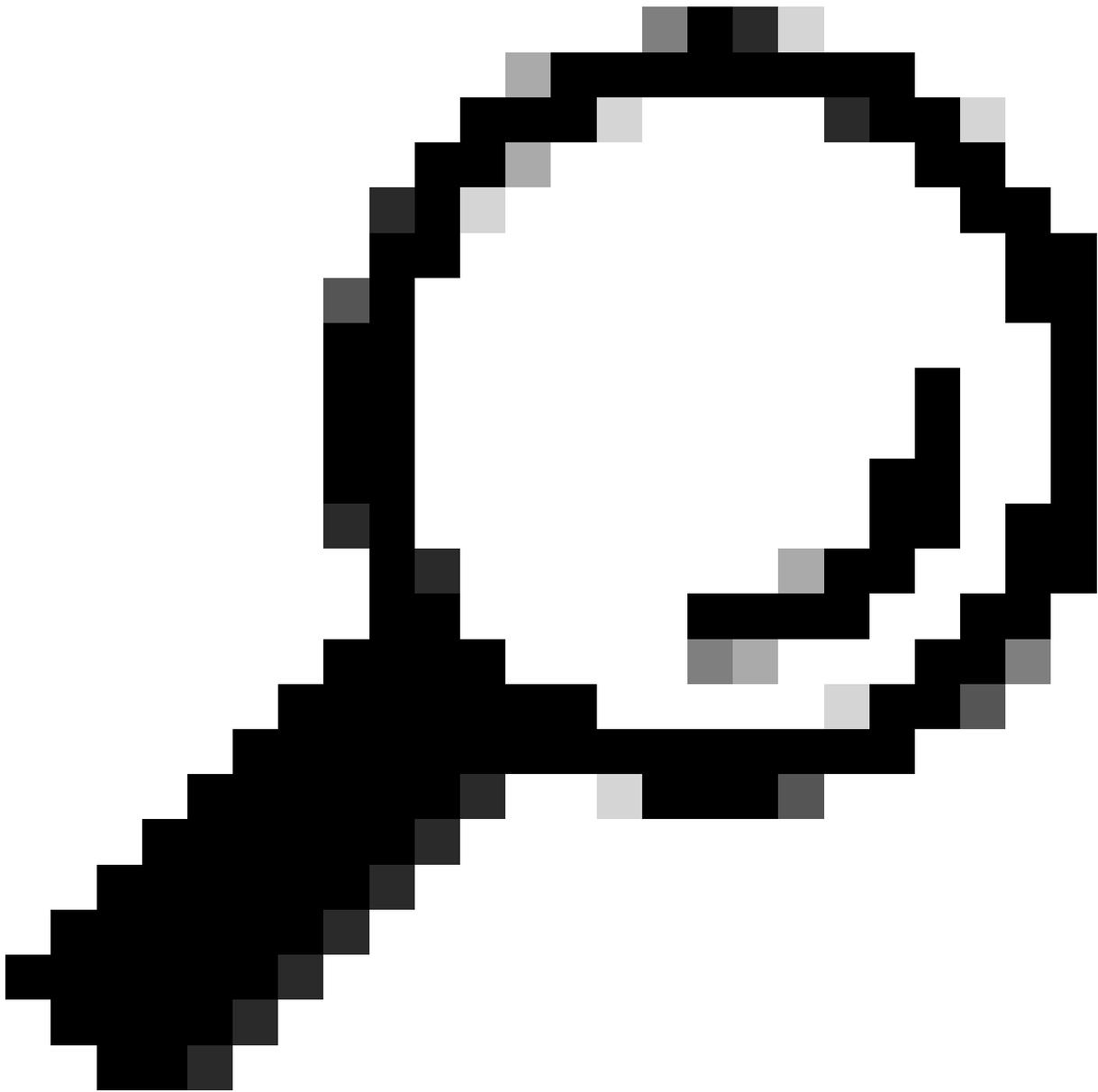
**reltime:** período de tiempo relativo, especificado como minutos | horas | días | semanas | valor de meses

**abstime:** periodo de tiempo absoluto, especificado como hh:mm:MM/DD/YY hh:mm:MM/DD/YY

**match:** coincide con una cadena determinada del nombre de archivo, especificada como valor de cadena

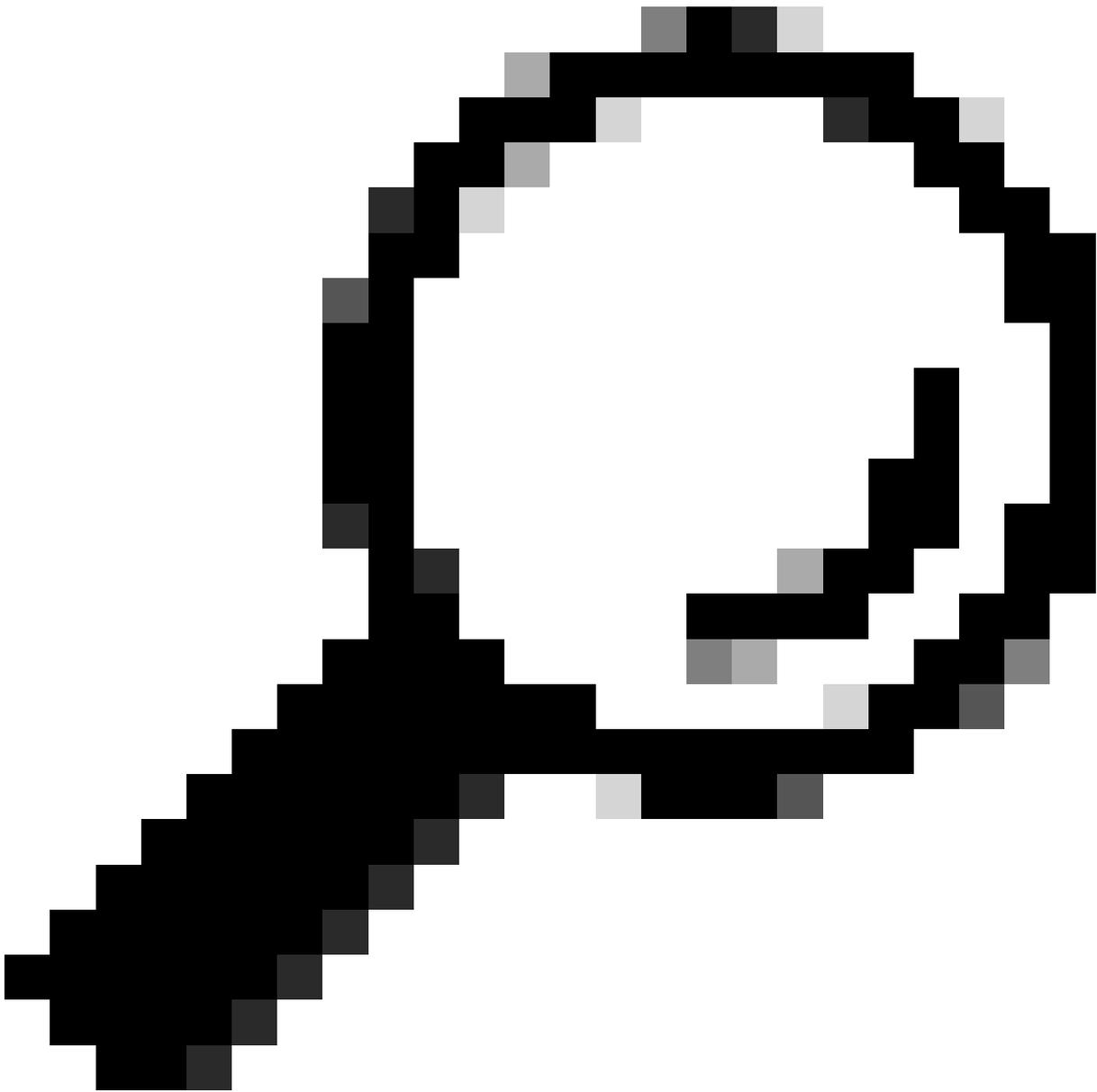
**recurs:** obtiene todos los archivos, incluye subdirectorios

**compress** le permite descargar los archivos en un formato comprimido.



**Sugerencia:** para descargar los archivos, asegúrese de que el servidor SFTP (protocolo seguro de transferencia de archivos) externo esté configurado y sea accesible.

---



**Sugerencia:** La opción `recurs` permite recorrer el directorio para todos los subdirectorios y archivos. Se utiliza si desea extraer todos los registros de un directorio.

---

#### Enlaces de referencia

- [Guía para desarrolladores de JTAPI](#)

- [Niveles de seguimiento de UCCX](#)

## Acerca de esta traducción

Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).